CMIF
Center for Multisource Information Fusion

Framework for AFFTC T&E of Information Fusion and Aerospace Vehicle Management Systems

Dr. James Llinas and Dr. Christopher Bowman

Report No. CMIF 1-01

February 2001

Acknowledgement: This research project was supported by Grant F49620-99-1-0090 from the Air Force Office of Scientific Research (AFOSR) and technical direction was provided by staff from the Air Force Flight Test Center (AFFTC) at Edwards AFB, California; the Center for Multisource Information Fusion is grateful to both agencies and their staffs for the support and guidance associated with this project.

CUB

CALSPAN—UNIVERSITY AT BUFFALO RESEARCH CENTER. INC.

UB University at Buffalo
The State University of New York

REPORT DOCUMENTATION PAGE

AFRL-SR-BL-TR-01-

0385

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
February 2001

3. REPORT TYPE AND DATES COVERED
Final Report (1/1/99 – 12/31/00)

4. TITLE AND SUBTITLE
Framework for AFFTC T&E of Information Fusion and Aerospace Vehicle Management Systems

5. AUTHORS
Dr. James Llinas and Dr. Christopher Bowman

6. FUNDING NUMBERS
G – F4962—99-1-0090

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Calspan-UB Research Center, Inc.
P.O. Box 400
4455 Genesee Street
Buffalo, New York 14225

8. PERFORMING ORGANIZATION REPORT NUMBER
CMIF 1-01 ; CMIF 2-01 ; CMIF 3-01

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Office of Scientific Research
801 N. Randolph Street, Room 732
Arlington, VA 22203-1977

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
None

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Publicly releasable; unlimited distribution

AIR FORCE OFFICE OF SCIENTIFIC RESEARCH (AFOSR)
NOTICE OF TRANSMITTAL DTIC. THIS TECHNICAL REPORT
HAS BEEN REVIEWED AND IS APPROVED FOR PUBLIC RELEASE
LAW AFR 190-12. DISTRIBUTION IS UNLIMITED.

13. ABSTRACT *(Maximum 200 words)*

The research objective of this grant is to provide understanding about the nature of multi-platform and distributed data fusion and the influence that such methods might have on flight-testing of future multi-platform systems at major range facilities such as, in particular, Edwards Air Force Base (the Air Force Flight Test Center, AFFTC), and also with a special focus on Electronic Warfare (EW) aspects and impacts. This report describes a basis and structure for both understanding and describing the Data Fusion-related issues and components of test operations in ways that are considered both technically correct from a Data Fusion viewpoint, and consistent from an architectural viewpoint. This instrument's purpose is to establish a consistent basis for contemplating and understanding any possible future test environment that involves Data Fusion processing concepts in order to cost-effectively define, design, implement, and maximally reuse test support components and data analysis capabilities at AFFTC; said otherwise, the bottom-line benefit of a Framework is affordability and efficiency in test operations and analyses. Our approach involves, for example, first determining the "role" for Data Fusion, and ultimately determining the design of Data Fusion components and detailed elements. As for any systems-engineering process, an important first step is to determine also the boundaries of the processes and functions to be addressed: what is inside the boundary of consideration and what is not; the items inside the boundary are labeled the "Black Box" components in this report. Also, important to the overall Framework definition is the process by which the Framework will be updated; we suggest formalized Configuration Control techniques as used for evolving software. This is because the test concepts and requirements for AFFTC will no doubt change beyond what can be envisioned today; thus, while we argue for a consistent and persistent approach to understanding, we nevertheless recognize that things change over time.

14. SUBJECT TERMS
Distributed Data Fusion, Flight Testing, Flight Test Planning, Flight Test Resources, Research Frameworks

15. NUMBER OF PAGES

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
U	U	U	UL

Table of Contents

List of Tables and Figures

Framework for AFFTC T&E of Information Fusion and Aerospace Vehicle Management Systems

Dr. James Llinas and Dr. Christopher Bowman

Abstract

This report summarizes the research conducted at the Center for Multisource Information Fusion (CMIF) at the State University of New York at Buffalo (SUNY at Buffalo) during the second year of a two-year Air Force Office of Scientific Research (AFOSR)-funded research grant. The overarching research objective of this grant is to provide understanding about the nature of multi-platform and distributed data fusion and the influence that such methods might have on flight-testing of future multi-platform systems at major range facilities such as, in particular, Edwards Air Force Base (the Air Force Flight Test Center, AFFTC), and also with a special focus on Electronic Warfare (EW) aspects and impacts. In this second year, the research has been entirely of a study type, involving a degree of familiarization of the university team with EW technology and techniques, and with research into the concepts for representing the complex information environments associated with multi-platform and distributed data fusion processing.

It should first be noted that this Framework report should be considered as a "model" or skeleton of an actual, complete Framework document, and that only AFFTC will have the authority to develop a truly-representative and complete Framework document for its own purposes. What has been done herein is to describe a basis and structure for both understanding and describing the Data Fusion-related issues and components of test operations in ways that are considered both technically correct from a Data Fusion viewpoint, and consistent from an architectural viewpoint. The motivation for this work in Year 2 resulted in part from discussions with AFFTC staff at the end of Year 1, during which the value of a Framework, in the face of the varied and complex future test requirements and operations facing AFFTC, was realized. This instrument's purpose is to establish a consistent basis for contemplating and understanding any possible future test environment that involves Data Fusion processing concepts in order to cost-effectively define, design, implement, and maximally reuse test support components and data analysis capabilities at AFFTC; said otherwise, the bottom-line benefit of a Framework is affordability and efficiency in test operations and analyses.

In carrying out the formation of this "model" Framework, we followed the approach that one would take to designing an actual Data Fusion process. This involves for example first determining the "role" for Data Fusion, and ultimately determining the design of Data Fusion components and detailed elements. As for any systems-engineering process, an important first step is to determine also the boundaries of the processes and functions to be addressed: what is inside the boundary of consideration and what is not; the items inside the boundary are labeled the "Black Box" components in this report. Also, important to the overall Framework definition is the process by which the Framework will be updated; we suggest formalized Configuration Control techniques as used for evolving software. This is because the test concepts and requirements for AFFTC will no doubt change beyond what can be envisioned today; thus, while we argue for a consistent and persistent approach to understanding, we nevertheless recognize that things change over time.

The logical next step in the progression of this Framework of understanding is to apply this prototypical thinking to real test case studies for future experiments planned for AFFTC; presuming continuity of this project, we see this as an important Year 3 activity.

1.0 Introduction

This report is about the second phase of a two-year effort to study, characterize, define, and prototype methods for the Test and Evaluation (T & E) of distributed data fusion systems. However, it is focused on the associated T & E implications for the major test range community and in particular Edwards Air Force Base in California. The effort stems from the visions for future combat depicted in various DoD forward-looking documents such as Joint Visions 2010 and 2020 (JV2010, JV2020), the Advanced Battlespace Information System (ABIS), Joint Battlespace Infosphere (JBI), and New World Vistas (NWV), among other similar reports. In those documents, sensibly all views of the future theater environment show a highly distributed but highly connected information environment, with the backbone data linking infrastructure generally labeled as the "Infosphere" or "Cybersphere". The gist is that such thinking also applies to the various platforms in the theater, including of course air platforms both for Precision Engagement and Intelligence, Surveillance, and Reconnaissance (ISR) purposes.

Perspectives from the Pentagon also echo these views; recent briefings by senior DoD officials describe these and other motivations for modernization and investing in the T & E infrastructure. In [Gehrig, 99], the challenges for OT&E are described as revolving around the *testing of systems-of-systems*; it could be argued that this view is an extension of the idea of T&E of distributed data fusion systems, at least in the sense of distributed data fusion systems as *informational* systems-of-systems. As stated in [Gehrig, 99], part of the testing focus will be on *interdependencies* among systems—again, this interdependency can be considered as an extension of the requirement to test inter-platform informational dependencies. Gehrig also shows that the OT&E workload (number of test projects) has been increasing for AFOTEC since about GFY1993, yet funding and manpower are reducing. These pressures result in a significant demand for modernization of the remaining T&E infrastructure required to support future acquisition programs. Key to the project at hand is the depiction of these future acquisition programs as involving, among other things, from [Gehrig, 99]:

- "advanced sensors
- real-time data processing
- massive comms and data handling
- advanced aircraft and munitions"

In particular, Gehrig also lists the "capabilities needed for Joint Vision 2010 initiatives", which includes: "large scale C4ISR systems testing", "threat-representative targets with multispectral signatures for realistic test conditions", and "information warfare technologies testing". Such systems certainly include modern air platforms and platform groups as well as the associated sensor processing, and these requirements characterizations are synonymous with the data fusion processing operations so central to the successful employment of these platforms.

Thus, the research conducted herein can provide part of the basic knowledge necessary to examine the issues, techniques, architectures, test plans and configurations for a variety of flight tests related to the following mission concepts:

- multiple sensor platforms feeding any centralized fusion node
- multiple UAV-enhanced surveillance (multiple UAV's + surveillance platform data fusion)
- "sensor-to-shooter" concepts involving onboard + offboard data fusion
- research on either Distributed and/or Intelligent Mission Controller concepts
- research in scaling 1 or n-platform flight test performance/results to N- platform configurations
- "leader-follower" concepts for interceptor systems
- combined sensing, fusion, and C3 between and among ground and air platforms

In discussing the role of modeling and simulation in OT&E, Gray [Gray, 98] asserts that AFOTEC must implement a "mission- level evaluation Framework" and to "measure effectiveness as a component of total force mix". These remarks imply that metrics and measures in T&E must shift to the mission-effectiveness level of definition. This is harder to do than measuring functional-level performance as previously done, since the effects of variables between the functional and mission-levels must be accounted for. As will be noted below, these factors push the testing focus away from DT&E toward OT&E. However, these demands also have implications for the testing of the distributed fusion processes between single or multiple attack platforms, supporting ISR platforms, and ground systems. Reference to the operational concepts for the F-22 and Joint Strike Fighter (JSF) immediately reveals the criticality of both the central and supporting information processing operations and information products (more is said on this below). Gray's description of the role for mission-level simulations for AFOTEC can be equivalently applied to the role for modern T&E, e.g.

- "force size / composition tradeoffs for mission accomplishment
- identify previously unknown capabilities and limitations of multiple-sensor configurations, among other factors.

Additionally, we can expect a future mission environment that is considerably broader[1] and that has the following features:

- Small Regional Conflicts
- Advanced Soviet Equipment
- Multi-spectral Acquisition and Tracking
  –RF, IR, UV
  –Multi-modal
- Non-traditional Tactics

among other factors". The *informational needs* in any missions of this type are quite broad but we have attempted develop a some representative categorization of these needs, defining situational awareness, lethality assessments, pilot alerts, and response management as a set of initial categories within which to examine and engineer the role of information. These have a bias toward and are limited to a focus on EW and IW operations. These categories are shown in Table 1.1.

[1] With some 13 OOTW (Operations Other Than War), 3 AT (Asymmetric Threat) and 9 "Gray Area" missions, one can easily develop a list of some 25 mission types in addition to those for conventional warfare operations !

In these future mission scenarios, and with the variety of theater ISR systems and platforms described above, these informational needs will be satisfied by a *complex, interdependent network* of these systems and platforms, involving distributed data fusion and dynamic resource management. This interdependency means that the utility and value of single-platform DT&E-and functional-specification-oriented testing will be minimal. This is not to say that such testing is unnecessary but that its role and contribution to overall system and capability development will be declining over time.

Table 1.1. Representative Informational Needs Categories

Situational Awareness Needs	Lethality	Pilot Alerts	Response Management
Offensive: –Targeting-oriented **Id/Track:** –Space-Spectrum/ Respond	**Factors:** –IFF Confidence –Aspect Angle –Range –Altitude –Operational State/Mode –With or Without CM's	**Mission Critical Events:** –Mode Changes –Pop-up Threats –CM Status –System Status –Maneuver Cues	**Response Inhibit:** –Threat requires CM •*Auto Mode*—above TH •*Covert Mode*—CM inhibit ex. those lethal –Threat does not require CM •*Warning-only Mode*
Defensive: –Survivability-oriented **Detect/ID/Track/ Status re DecMkg / Disrupt:** –Vulnerability			**Response Selection:** •CM Assignment –Vs Threat System State –Priority-based •Expendables –ECM(RF,IR)/Maneuver/External –Optimality Trade
Commander's Catechism: –Where is he –What is he doing –Going to do –How many –How to respond –etc	--Location --Behavior --Predictive capability --Order of battle	--Current, predicted behavior	**Resource Control:** •*Jammer Control* –Pause/Inhibit/Enable •*Expendables Control* –Type/Technique •*Active/Cued/Hold* •*Priority* •*Maneuver Control* –Time of maneuver –Coord w CM •*Weapon Cues* –Onboard Fire Control –Offboard weapon system
Threat System Creation: –Emitters –Spatial Correlation –A Priori Data	**System State:** –Search-Acquisition-Track-Missile Launch –IFF Assessment **EOB Correlation/ Threat System Type/IFF Replies:**	**State Prediction:** –Position (Mobiles)/ **Mode/ Time-Range-Lethality:** –TTGo	

Instead, what will be needed is a new, flexible and affordable T&E infrastructure for testing and evaluating these system-of-systems environments; flexibility and affordability of that infrastructure will be achieved in part by an infrastructure "Framework" that is in effect reusable, as a result of an infrastructure design that is based on understanding the functional and processing commonalities across missions, multi-platform systems, and concepts of employment. This Framework document is a first step toward achieving that goal.

References
[Gehrig, 99] Gehrig, J., "DoD Perspectives on Test and Evaluation", presented to the TECOM Test Technology Symposium, Mar 1999.
[Gray, 98] Gray, F., "Applications of Modeling and Simulation to Operational Testing", presented at the TECOM Test Technology Symposium, Johns Hopkins APL, March 1998.

2.0 The Changing T&E Context for AFFTC: Motivation for a Framework Document

The Introduction has given some background on the changing environment for testing and evaluating modern-day combat systems and platforms. In this section we elaborate further on this theme, which we call the "context" for T&E. Our purpose is to establish the rationale or motivation for the major content of this document, which we call a "Framework" for T&E at AFFTC. A "Framework" is a *mechanism or structure* to define and document the needs-driven and role-constrained interrelationships :

(1) *between* AFFTC and external systems, services, customer I/F's, data links, and
(2) *among* internal AFFTC T&E functional components

The purpose for directing a major portion of second-year effort toward the formulation of this Framework is that the Framework, in our opinion, establishes a basis of understanding of the complex and broad new context for T&E that will lead to: (1) improved affordability of T&E activities at AFFTC, (2) improved understanding of the role and nature of data fusion processes and technologies in modern-day T&E environments, and very importantly, (3) a perspective (a structure) within which all (or at least most) future T&E activities can be viewed consistently and in a modular fashion. The Framework will provide the means to:

- *describe* the role for software (SW) and hardware (HW) test articles hierarchically from concept modules to full systems
- *support* test progression and levels of abstraction in testing
- *represent* alternative stimulations, simulations, avionics test articles, effectors, HIL, and performance evaluation approaches for testing
- *define* the structure of the fusion and management avionics testing components, interfaces, and application of the Framework
- *support* affordable performance evaluation of avionics test articles
- *enable* representation of the role for all projected tests of aerospace vehicle software (e.g., data fusion and resource management) and hardware (e.g., sensors and countermeasures)
- *support* 412th Test Wing Preliminary and Detailed Capabilities Assessments
- *support* the representation of the test progression (e.g., what should be simulated, real time, real data, HIL, and flight tested) for each test article
- *supply* an applications layer architecture for data fusion and resource management that conforms to standard open layered architectures (e.g., GCCS)
- *provide* a performance analysis methodology to reveal fusion and resource management performance as part of a distributed network

The Framework shall be applicable to essentially all data fusion and resource management testing applications, testing of avionics concepts through mature systems, and from 1-on-1 vehicle subsystem testing to m-on-n battlespace management testing. The need for the Framework was in part a result of our first-year's research, which showed that there are many changes and technology challenges that can be expected in future range T&E activities, and that a consistent top-down view of this dynamic and complex landscape was needed. But this was not our own view; at a Technical Interchange meeting (TIM) on March 8th 2000, held at the Center for Multisource Information Fusion (CMIF) at the State University of New York at Buffalo (SUNY@Buffalo), staff from AFFTC agreed to this need, during the course of a briefing on the Framework concept.

A Framework is needed to describe the *role* for each test of air vehicle functions (i.e., data fusion and resource management) and HW (i.e., avionics, sensors, data links) within the test environment provided by AFFTC. This role needs to be described within a common Framework so that it can be repeatedly applied to the testing of AF programs from concept development to mature aerospace vehicle systems. The primary components of this Framework should include the scenario stimulators, sensor simulators, avionics (i.e., not covered by the test article itself), effectors, users, and performance evaluators. The goals of the common Framework for AFFTC testing include the following:

- permit achievement of useful results while minimizing costs
- facilitate user understanding and communication
- permit comparison and integration
- promote expandability, modularity, and reusability

The Framework is further needed to support 412th Test Wing Capability and Approach assessments. This includes the development of an AFFTC testing system concept (e.g., architecture) and how it maps to AF programs. Moreover, this analysis Framework needs to support the determination of the test progression (e.g., what should be simulated, real time, real data, HIL, and flight tested?) for each test article.

The Framework also needs to contain a performance analysis methodology to reveal fusion and resource management performance as part of a distributed network. This methodology needs to be applicable to the testing of AF avionics concepts through fully developed systems and from 1 on 1 vehicle subsystem testing to m-on-n battlespace management testing.

The Framework architecture needs to define the *structure* of the fusion and resource-management avionics testing components, their relationships, and the principles and guidelines governing their design and evolution over time. Furthermore, the Framework needs to apply to nearly all data fusion and resource management testing applications. Standard open layered architectures already exist below the applications program interface (API) (e.g., GCCS, TBMCS, JMCIS, etc.). Thus the need at AFFTC is for an applications layer architecture for data fusion and resource management testing. This applications layer architecture needs to provide a canonical functional partitioning which is upgradable and reusable to include the following:

1. *Levels of Hierarchy* - the architecture should be able to accommodate alternative design approaches. Therefore metrics and interfaces should be established at each level so that system

designers with alternative techniques could replace a function object and still interact with the rest of the architecture.

2. *Levels of abstraction* - partitioning the processes in such a fashion that the information is consistently abstracted as it goes from the lower to upper objects.
3. *Balance of Breath vs. Depth* - objects should be defined in such a way as to minimize possible bottlenecks i.e. where there is a lot of depth of knowledge, minimize breath; higher level objects have more breath less depth.
4. *Object-oriented modular design* - with common functional objects using inheritance and standard interfaces for information fusion and resource management components.

In summary, the requirements for the fusion and resource-management testing Framework include the following:

- Describe the role for test SW and HW test articles hierarchically from concept modules to full systems
- Support test progression and levels of abstraction in testing
- Represent alternative stimulations, simulations, avionics test articles, effectors, HIL, and performance evaluation approaches for testing
- Define the structure of the fusion and management avionics testing components, their relationships, and the principles and guidelines governing their design and evolution over time so as to support affordable performance evaluation of avionics test articles.

We emphasize that this particular Framework document will ***not*** address and satisfy all or even most of these goals and requirements for a Framework; to do so is a major undertaking beyond the scope of this university research task. The goal of this document is to characterize the issues and approach methodology for achieving a Framework *for information fusion and related resource management functions in particular*; it must be noted that there are many other issues and functions that will need to be addressed in establishing the fully-comprehensive Framework. In what follows, we suggest a "spiral" development approach toward defining and enabling the Framework; this document, for the fusion and resource-management functions mentioned, is the first such spiral for those functions.

3.0 The Need for a Framework Development Methodology

It is all-well and good to argue the rationale for the Framework itself but a critical question is: How shall that Framework be developed? Clearly some methodological approach is required that is orderly and complete. The methodology suggested herein is, by and large, derived from the so-called "spiral" method of engineering information systems. Compared to other approaches, this approach has high flexibility in its ability to incorporate yet other methodologies within it (e.g. waterfall model, which has the benefits of increased control and accountability), but it demands proactive managerial decision-making within each spiral. By and large, its main strengths derive from its orientation toward reevaluation of design perspectives, exploitation of technology, and risk management; as a result, many experiences have shown it to be well-suited to complex, dynamic, and innovative projects.

Figure 3.1 shows the typical graphic depiction of the process; its main steps are, in terms of defining a Framework; see [Boehm, 86]:

- Define Framework objectives, constraints
- Identify Framework alternatives
- Evaluate alternatives with respect to risk
- Develop, verify next version of Framework
- Determine methodological approach to next spiral

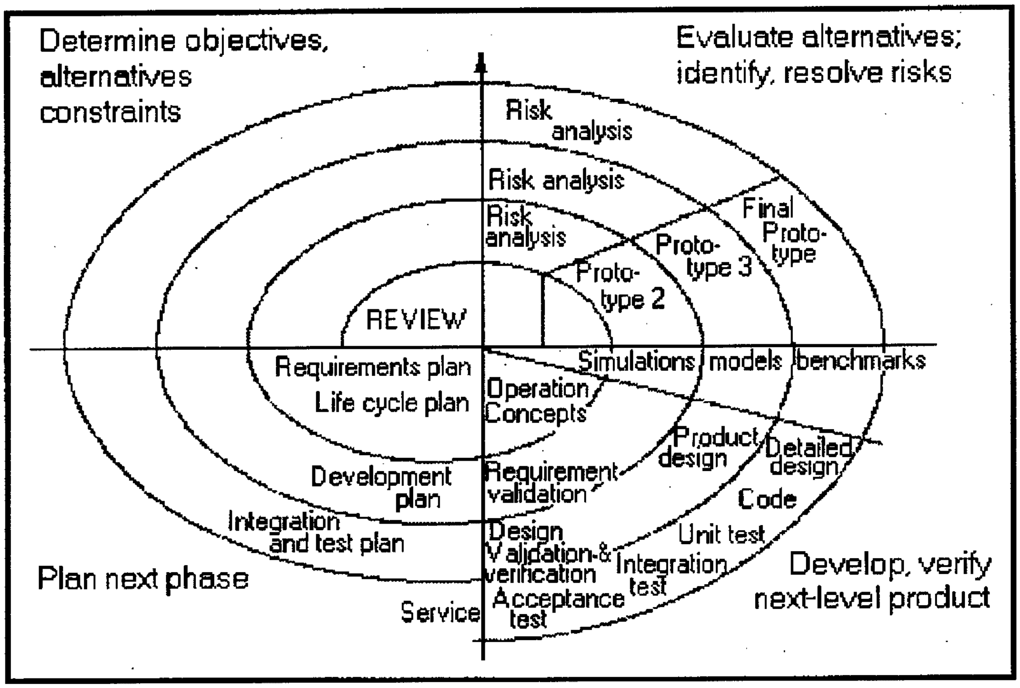

Figure 3.1. Typical Spiral Development Process Characterization (from [Boehm, 86])

A method similar in many ways to the spiral approach has been used to develop Data Fusion processes; this method has been known as the "Data Fusion and Resource Management Tree" or DF&RMT approach [Bowman, 94]. More recently this has been called the "Dual Node Network" or DNN approach to architecting DF processes. Figure 3.2 shows a top-level diagram depicting this method, where the ellipse in the upper box (and applicable to each box) shows the "spiral" process notion between the phases shown. The notion of risk is not as overtly evident but implied in the "Performance Analysis" box. Note the important distinction at the moment that the process of Figure 3.2 is for DF process design, not Framework design. If we carefully examine and modify this figure for our Framework purposes, then we will have a representation of the same methodology but as applied to Framework design and modification; this is shown in Figure 3.3.

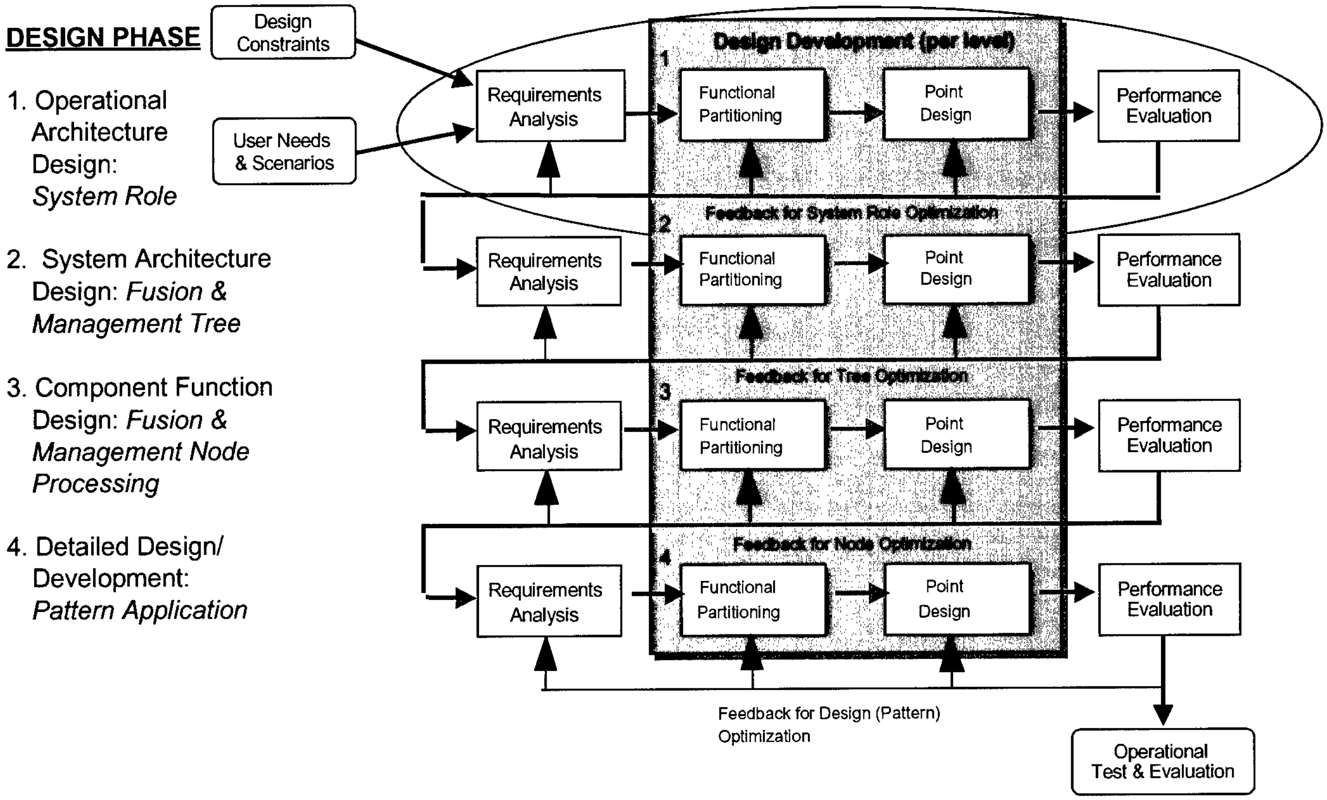

Figure 3.2. Top-Level Depiction of DNN Approach to Data Fusion Architecture Development (From [Bowman,94])

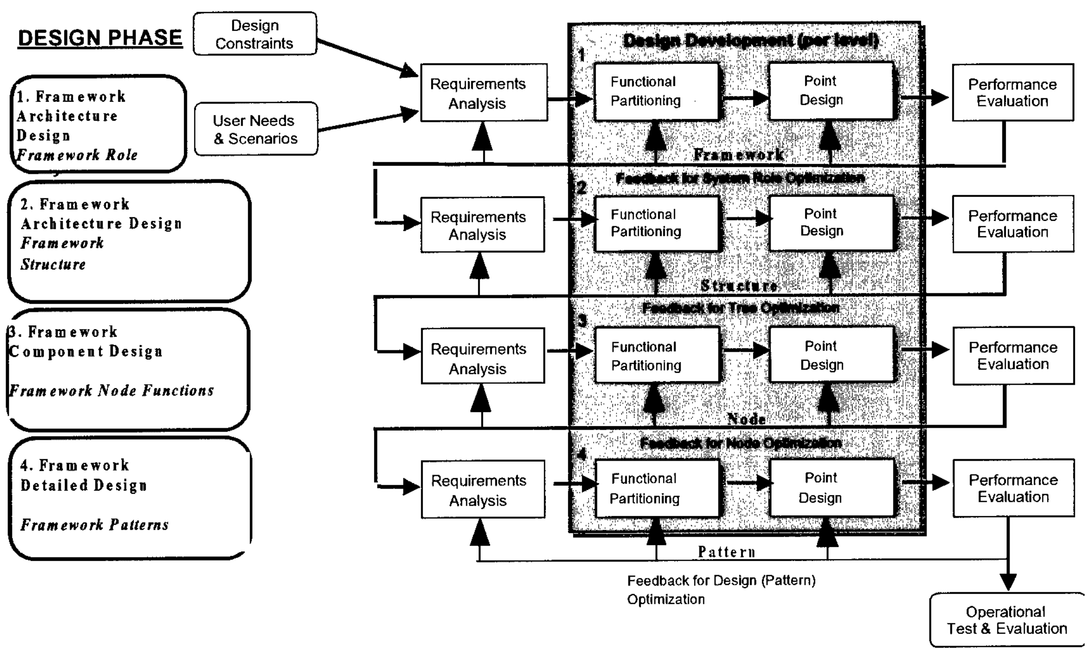

Figure 3.3. DNN Approach Modified for Applicability to Framework Architecture Definition

It is very important to understand what are rather subtle differences in these two figures. The Framework is not a fusion process or any process at all; it is a way of organizing AFFTC's view of the structure, interrelationships, and organization of what AFFTC defines as relevant to the T&E of all future test programs that involve data fusion technology. As we have said above, without this complete and consistent view of this portion of AFFTC's testing landscape, inefficiencies, errors, and misinterpretations of those test programs will be likely. Note too, as we have said in Section 2, that we emphasize that this report is *not* "the" Framework but a description of a methodology to determine such a Framework, with some examples and recommendations included. It will be *AFFTC* that ultimately determines the details of *the* Framework as applicable to their future test programs. (The CMIF proposal for CY 2001 however does suggest that the next steps should involve CMIF working in conjunction with AFFTC to finalize the first version (spiral-version) of this Framework.)

References

[Boehm, 86] Boehm, "A Spiral Model of Software Development and Enhancement," in ACM Software Engineering Notes, August 1986, pp.14-24.
[Bowman, 94] Bowman, "The Data Fusion Tree Paradigm and It's Dual," Proceedings of 7th National Symposium on Sensor Fusion, invited paper, Sandia Labs, NM, March 1994.

4.0 Moving Toward OT&E

As mentioned previously, in the face of these changing mission types and ever-broadening informational requirements, it is considered that another major change in the context of range T&E is a movement from what has historically been developmental T&E (DT&E) to operational T&E (OT&E). It is not only the changing missions and informational spectrum that suggest this view but also the *interdependencies* of functions and platforms within the concepts of employment for new tactical platforms. Given these interdependencies, the only way the DT&E can be carried out is under highly constrained conditions, with all external interdependencies held constant, in a highly "conditional" T&E approach, and one that portends high cost for relatively little T&E product (i.e. knowledge). It would appear that since concepts of employment define the interdependencies, and since the context of those interdependencies must be provided by the T&E organization or facility, it would be much more cost-effective to carry out OT&E since the expense of providing some version of the operational environment has been incurred in any case.

4.1 Four Critical Questions

Whether AFFTC does indeed move toward OT&E or not, there will be in any future test project four critical questions to deal with:

How many aircraft (real or virtual) will be involved in future testing?

- What functions will be tested?
- What is the role for fusion, meaning Red-White-Blue fusion?
- What are the concepts for the "test articles"?

The first question relates to AFFTC's long heritage of single-aircraft-based DT&E. With the employment concepts for future tactical aircraft such as JSF and F-22 depicting the interplatform dependencies we have been discussing, there is a question of how those multiplatform environments will be provided in the test environment, and whether multiple full-scale aircraft will be employed on any given experiment. Clearly these are significant cost-driver issues that must be dealt with. The second question deals with the other-than-EW functions that will be involved in such tests; with AFFTC having a heritage of EW-focused testing, extending the functional repertoire (driven in part by the involvement of data fusion processes) will cause new methods, metrics, and test facility type requirements to be addressed. The third question addresses the nature and extent of the employment of data fusion technologies in any given experiment. It is of course assumed that "Blue" or friendly-system data fusion will be involved and of interest (in essence as an element or function of the "test article"), but there will be questions as to whether and how "Red" or hostile data fusion functionality will be provided, since even current-day hostile systems employ data fusion techniques (see Appendix B below). Finally, there is the potential to employ data fusion techniques for range testing support purposes ("White" data fusion), for example to fuse multiple range sensor data to develop improved estimates of true platform locations for post-test analysis. The last question addresses the challenge of defining the "test article" physical and functional boundary; for example, the Blue data fusion processes could be within the test article boundary, as could platform tactics, countermeasure techniques, etc. This question asks: "what exactly is being tested?", which, in the new mission concepts of the future, will encompass much more than a single EW device, for example, and is expected to include multiple, layered, distributed Blue data fusion processes, certain operational tactics, and EW and IW devices and techniques, for example.

5.0 AFFTC Test Framework Role –Phase 1

At this point we begin a description of the Framework methodology; following the depiction in Section 3, we begin by considering the *role* for the Framework, meaning the *T&E domain over which the Framework will be applicable or pertinent to.*

5.1 AFFTC Test Framework Requirements[2]

In this first step, when we discuss "requirements" we mean those functional requirements that will be pertinent to or for which the Framework will apply. The very first analyses for this step are those in which AFFTC will take a careful look to future test programs at their various stages of development, as well as future aircraft and associated C2 concepts of employment. Another way of describing the goal of this role-defining step is that it is directed to defining the "problem space" for future T&E operations that AFFTC will or would like to participate in.

It is expected that AFFTC will need to conduct analyses of:

--mission space
--function space
--platform space
--geographic space
--etc

[2] The reader should note right at this early point in this document that the requirements stated below are suggested, first-cut requirements. As stated previously in Section 3, the entirety of this document should be considered as a preliminary view of a rather complex process, ultimately to be carried out in detail by AFFTC.

by way of developing this initial view. Next steps will involve partitioning decisions in these spaces to define what is included and what is excluded from AFFTC's definition of what it considers its T&E operational space to be. In conducting such partitioning analyses, it is usually healthier to take an "excluding" or critical approach, being aggressive in minimizing, with insightful judgment of course, what is excluded from the operating space. At this point in the analysis, an organized enumeration of the following elements will have been defined:

--Pertinent Missions
--Pertinent Functions
--Pertinent Platforms
--Pertinent Geographies
--Pertinent Customer Organizations
-etc

--All functions that are pertinent to AFFTC T&E
--BOTH internal to AFFTC
--AND immediately external to AFFTC

The above suggestions for a requirements analysis closely follows the approach described in [Wirfel, 00], who describes:

- Platforms, capabilities, systems
- Emerging EW technology and techniques

This briefing by Wirfel also includes thoughts about EW related flight testing and test support requirements, which we will review here. In terms of platforms, Wirfel lists:

- Manned aircraft
- UAV's
- UCAV's
- Satellites

He suggests that Manned Aircraft will still be the focus in the near term. These platforms will have integrated and automatic sensors, weapon systems, and countermeasures. He suggests that the EW test emphasis will be on survivability, and sensor and weapon effectiveness. Interestingly, he does not include data fusion in this discussion. He asserts that UAV's and UCAV's will be the primary future sensor platforms, for which "significant EW test opportunities" will exist. Understanding and predicting what these opportunities might be will require careful assessment of the concepts of employment for such platforms. It should be noted that the potential future use of such semi-autonomous platforms is what gave rise to our first-year research into the area of intelligent agents and multiple, distributed intelligent agents. Wirfel suggest that these platforms will be carrying out a broad range of functions, e.g.

- Surveillance/Reconnaissance
- Identification

- Targeting
- Electronic Attack/Protection
- Suppression of Enemy Air Defense
- Sensor-to-Shooter
- Weapons Employment
- Battle Damage Assessment
- Mine/NBC detection

Wirfel also suggests that space-based platforms will play a variety of roles as well, listing the following functions:

- Surveillance / Reconnaissance (primarily SAR/MTI)
- Combat ID
- Sensor-to-Shooter
- Battle Damage Assessment
- Communications
- Use of GPS

Another remark made in [Wirfel, 00] is that one future significant threat is the coherent jammer, which will generate requirements for support sensor platforms to monitor the threat (since platform radars, according to these assertions, will be "useless"). In a similar line of discussion, Wirfel suggests that advanced decoy/false-target generation systems (Canadian "CARDS" system is mentioned) will also require support sensor systems.

In summary, it is analyses of this type that AFFTC must conduct to develop a vision of the future role for AFFTC and, as regards this particular Framework we are discussing here, the Data Fusion thread throughout this overall vision.

Following Wirfel, for example, the near-term emphasis would be on Manned Aircraft and advanced EW systems and components such as coherent jammers and advanced decoy/false-target generators, etc. The mid-term focus would be on preparations for T&E of Manned Aircraft working in conjunction with support sensors such as satellites or various ISR assets, and the far-term focus would be on UAV's and UCAV's.

5.2 Strawman Role for the AFFTC T&E Domain ("Partitioning and Black Box Design")

As information of the type described above is gathered and analyzed, it needs to be partitioned into a "core" set of functional concerns for AFFTC—we call this core, in this document, the *"black-box"* set of functions, using terminology borrowed from software system design. These functions at this point are not themselves partitioned or analyzed but simply collected and listed. These are the functions that AFFTC will both consider and/or provide as part of its T&E operations, and are "internal" to the black-box. In conjunction with this set of functions is another set which represent the functions immediately external to the black-box set, that is, those that are the tightly-coupled interfaces to each of the black-box functions. These external functions are also collected and listed but also partitioned, at least at a high-level in this first phase analysis. A first-cut depiction of this type of analysis is shown in Figure 5.1 below. Note that there are yet further external functions to this figure; that is, this figure is already a reduced depiction of the "problem space" that AFFTC will consider[3].

It can be appreciated then, that by "role" definition for the Framework, we mean a depiction of this type, showing those functions that will be central to *AFFTC's T&E operations for Data Fusion-capable test articles.* Related to those operations, we enumerate the black-box functions as shown in the example figure, and also the immediately-external functions. In a proper Framework document, these functions and their interfaces would be described at an adequate level of detail.

5.3 Evaluation of the Role for AFFTC Testing

As has been noted, Figure 5.1 should be considered a *representative* starting point in defining the role of a Framework for AFFTC testing in any given application. That is, it is acknowledged that certain test activities and programs will require some modification /feedback of the above "black box" boundaries; however, the position that AFFTC should take in the future, when this black box is finalized, is a relatively stiff one regarding changes to this black box, since the entire premise of the value of a Framework is its basis for standardization and the attendant benefits. Modification of these boundaries must be done on a cost-effectiveness basis, as AFFTC clearly must remain economically viable as a test range if it is to survive.

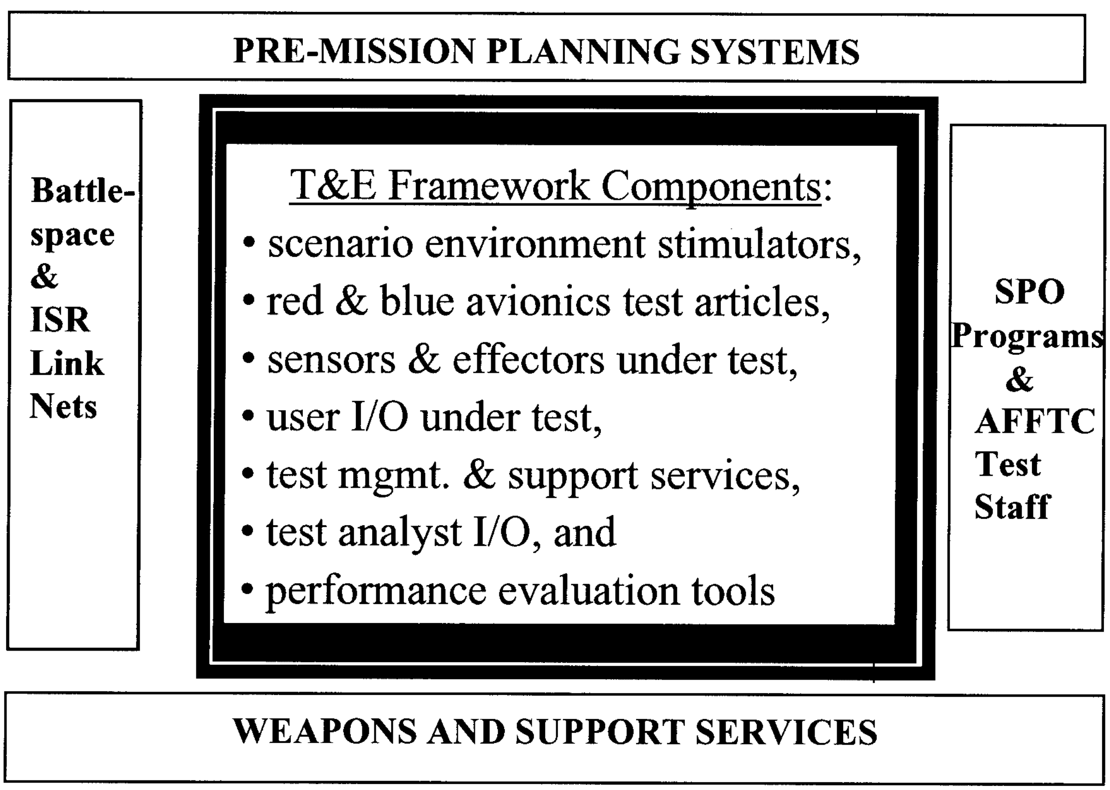

Figure 5.1. The Functional Role For AFFTC Avionics Testing

[3] Recall, once more, that this too is just an example, *not* the "recommended" role characterization for the Framework.

Three of the four external boundaries above could be called "technical" interfaces: weapons and support services, battlespace and ISR link nets, and pre-mission planning systems. The fourth interface, to SPO-based customer programs is, however, probably the most important in the strategic sense; this is because modifications to the technical components of AFFTC will generally be expensive but may be cost-effective in the long run if an AFFTC customer can commit to multi-year programs requiring those facilities. Another strategy for cost-effective black-box modification would be to determine if these external capabilities can be "out-sourced", i.e. provided as non-recurring cost elements of the range facilities. This could in fact be done through cooperative arrangements with other test ranges; e.g. weapons capabilities could be provided by Eglin or China Lake, etc. Further, there is the question of the level of fidelity at which these external capabilities need to be provided; if they can be provided in software or low-fidelity hardware, then this is another tradeoff approach, which also may be achieved through cooperative arrangements (e.g. SURVIAC software models).

References

[Wirfel, 00] Wirfel, J.A., "EW Test Topics", Briefing of February 11, 2000 provided to CMIF by Kurt Buehler of AFFTC.

6.0 AFFTC Testing Framework Component Design—Phase 2

Once the black-box boundaries have been determined, a generalized picture of the functions and levels of fidelity to be provided for each function for all test programs has been defined[4]. However, these Framework (or Black Box) functions at this point have been simply enumerated; their interrelationships and organization have not yet been defined, which is the purpose of this phase. To accomplish this, it is necessary to refine the requirements for the Framework (i.e. analyze them to greater detail) and develop "designs"[5] for the necessary components.

6.1 Black-Box Requirements Refinement

The above example of black-box (BB henceforth) functionality was derived from our first-cut assessment of the functions that AFFTC would likely be concerned with providing and dealing with in future test programs. Building upon this example, we develop an example of the *requirements refinement* for these functions in the following.

First we discuss the Scenario Environment. There are two aspects to enlarging on the requirements for this function: one is to define the range of scenarios of interest, and the other is to define the notion of "environment". If we think of the mid-term, representative scenarios of interest would relate to the Joint Strike Fighter, as one relevant example.

The JSF is of course a multirole fighter, and so it has a broad range of missions that it must be capable of. Except for Marine Corps missions, one common mission application is the SEAD mission. The SEAD mission can of course be executed both lethally and non-lethally, using either EW or IW

[4] The phrase "all test programs" is of course conditioned on the idea that the Framework document is a living document under version control by AFFTC.

[5] *Definitions* may be a better term, as another purpose of the Framework is to minimize specialized designs; the notion of defining the required components presumes that over time AFFTC would assemble an inventory of reusable components that can be integrated in a special configuration for any given test.

techniques accordingly. However, it is estimated that Strike and AI missions will maximally stress the functions associated with Onboard/Offboard Fusion and the onboard information management systems; in the Lockheed/Martin JSF prototype [see Joint Advanced Strike Technology report: On-Board/Off-Board Information Fusion and Management Study, Final Rpt, CDRL A003, Lockheed Martin Corp, Mar 1996], this system is the Advanced Information Management System or AIMS, shown in Figure 6.1.

The Strike and AI missions typically face medium to high threats during ingress, target area, and egress, and have the best potential to stress the offboard support and to provide the data necessary to define functional requirements for the AIMS and by implication the DF/RM functions.

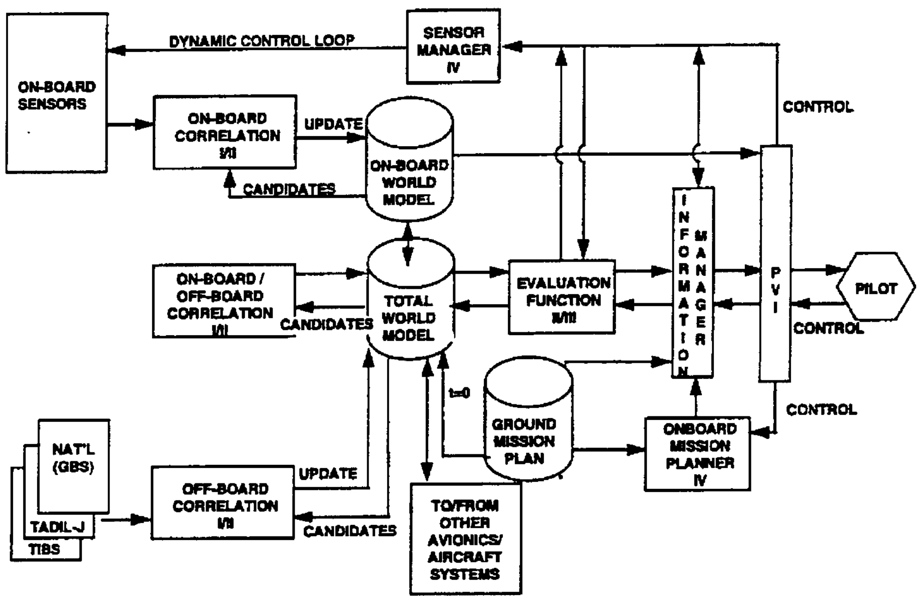

Figure 6.1. Preliminary Advanced Information Management System (AIMS) Functional Architecture for the JSF

In the far-term, again following Wirfel, AFFTC's focus may be on UAV's and UCAV's. We can look to DARPA programs today regarding these platforms to see what typical scenarios are like. In the case of UCAV's, DARPA has an ATD program (Joint DARPA/Air Force Unmanned Combat Air Vehicle (UCAV) Advanced Technology Demonstration (ATD) program) related to UCAV's that offers some ideas about scenarios. The operational UCAV system is envisioned as a force enabler that will conduct Suppression of Enemy Air Defense (SEAD) and strike missions in support of cost-2010 manned strike packages. The initial operational role for the UCAV is a "first day of the war" force enabler which complements a strike package by performing the SEAD mission. In this role, UCAVs accomplish preemptive destruction of sophisticated enemy integrated air defenses (IADs) in advance of the strike package, and enable low-risk operations by the attacking forces by providing reactive suppression against the remaining IADs. Throughout the remainder of the campaign, UCAVs provide continuous vigilance with an immediate lethal strike capability to prosecute high value and time critical targets. By effectively and affordably performing those missions, the UCAV system provides "no win" tactical deterrence against which an enemy's defenses would be ineffective, thereby ensuring air superiority.

Thus, we see the UCAV concept of employment as multi-purpose, ranging from weapon delivery to ISR, i.e. from preemptive destruction to reactive suppression to vigilance. Presumably, UCAV's would work in teams and would execute coordinated attacks on any IADS. Supporting test programs associated with these types of operations at AFFTC would require facilities not unlike those for JSF-type operations, although one can envision some new requirements—e.g., range safety when dealing with automated systems will need to be reviewed; similarly UCAV recovery techniques will have to be established; on the DF side, these systems will require automated DF processes having high reliability in the software sense, else software failures during test operations could invalidate part or all of the overall experiment; also, new methods of performance evaluation that condition the performance on the overall automated logic in the software as a whole will need to be established.

For scenario-type information on typical UAV's, we look to the Predator and Global Hawk UAV programs. By and large, these UAV's are intended for such applications as:

- surveillance
- near real time (NRT) targeting and precision strike support,
- NRT combat assessment,
- enemy order of battle (EOB) information,
- battle damage assessment (BDA),
- intelligence preparation of the battlefield (IPB),
- special operations support, and
- sensitive reconnaissance operations.

However, they could also be employed for atypical missions of the type mentioned above, such as treaty monitoring, blockade and quarantine monitoring, humanitarian aid, and disaster monitoring, among yet other possibilities.

The Medium Altitude Endurance Unmanned Aerial Vehicle (MAE UAV), Predator, provides affordable medium altitude reconnaissance and surveillance with a rapid deployment capability. Current national, theater, and tactical intelligence collection assets do not provide for long dwell, releasable near-real-time intelligence information on fixed and mobile targets for the in-theater CINC, Joint Force Command (JFC), and the National Command Authority (NCA).

Now consider the ***Blue Avionics Test Articles***, and Sensors Under Test for a UAV such as the Predator. The Predator is fully autonomous, low cost, and interoperable with current theater architectures. The Predator provides a near-term capability with potential cueing from satellites, Joint Surveillance Target Attack Radar System (Joint-STARS), U-2s, RIVET JOINT, and AWACS. The system takes advantage of available technology to provide continuous, near all-weather day/night coverage with EO/IR and SAR sensors and produces releasable/unclassified image products. The Predator can operate untethered and ground control is only needed for updating its activities. It is ideally suited for continuous observation over lightly defended areas when rapid deployment is necessary.

While Predator flies at moderate altitudes, Global Hawk flies at high altitudes and is intended to complement manned and national reconnaissance assets by providing continuous all-weather, wide-area, high-resolution imagery (EO, IR, and SAR) coverage in support of military operations. Global Hawk is to operate in low-to-moderate risk threat environments and is optimized to support those surveillance missions in which long range, extended endurance and long dwell over the target area are paramount.

According to DARPA information, for a notional mission, Global Hawk will have an operating radius of about 6,000 km, a loiter speed of 340 knots, an operating ceiling of 65,000 ft, and a maximum on-station endurance of 24 hours. Each sortie can undertake surveillance of a 136,900 km^2 area in the wide area search mode, while 1,900 spot targets can be prosecuted.

At the highest levels of its architecture, the Global Hawk system comprises three main segments; the air segment, the ground control segment, and the ground support element. The air segment consists of two primary elements: the Air Vehicle and its Sensor Payloads. The ground control segment consists of two primary elements: the Launch and Recovery Element (LRE), comprising a portable shelter for system health monitoring, and the Mission Control Element (MCE), which is a portable shelter that is responsible for key mission plan elements including flight, communications, sensor processing and aircraft and mission payload control, and can control up to three UAVs simultaneously. Finally, the ground support element includes all equipment required to operate and maintain the system, spare and repair parts, and personnel trained to maintain the air vehicles and ground elements.

The implications for AFFTC of preparing for and executing test programs on these platforms can be expected to be significant. Here again AFFTC will be dealing with *semi-autonomous systems* with all the range safety implications already mentioned for UCAV's. UAV's can again be expected to operate in groups and to have embedded teamwork logic in their software. This logic will be central in driving the platform and sensor operations in a mission-specific way, and will thus enter into the performance assessment process. Moreover, it is clear that UAV's are not likely to be operating on their own—at least not those of the Predator/Global Hawk variety, due to their vulnerability. (Note that both are described as operating against light to moderate defenses.) Thus, at least until more capable UAV's are developed it can be expected that they will operate in conjunction with UCAV's as an example, or possibly with advanced manned aircraft. Representing the true operational combinatorics of these multi-platform systems will be a major challenge for AFFTC. As for ***User I/O Under Test***, we can see from these systems that they are not fully autonomous but are subject to human control during flight operations (but these are today's systems, not the actual UAV's of tomorrow). In any case, in the same way as is done for today's manned aircraft tests, the "human factor" will have to be accounted for in evaluating test results. For UAV's, this human role could also be allocated to the ***Test Management and Support Services*** function, depending on how AFFTC will choose to look at this function. This is typical of the type of decisions that the Framework is in fact useful for.

Depending on how AFFTC views it, the ***Test Analyst I/O*** function could be quite proactive during the "runtime" of the test, or it could be more passive. This function for example would seem to have possible overlap with the human role for UAV Mission Control, or with Test Management. This is another example of how the Framework development process will aid (and force) AFFTC to deal with these decisions.

Finally, as we consider the ***Performance Evaluation Tools*** function, especially in light of the notion of moving toward OT&E versus DT&E, there is the critical question of performance versus effectiveness. AFFTC has a legacy primarily embedded in DT&E and performance evaluation. Movement to OT&E and mission effectiveness analysis is a major jump in both capability and viewpoint. Conducting effectiveness evaluations bears the burden of traceability—traceability through the chain of effects and factors that lead to, and influence, the final calculation of an effectiveness metric. Figure 6.2 shows this idea notionally, wherein the long chain of effects from signal detection to effectiveness metrics is shown.

This figure shows that, from a DF point of view, there are many factors between the evaluation of data fusion performed to enhance detection processing (e.g. to deal with stealthy hostile platforms), and the evaluation of, say, how layered data fusion processes contribute to probability of kill as a mission-effectiveness metric. Between those levels, as shown in the figure, are perhaps several DF processing operations such as DF-based target tracking or location, target identification based on DF, as well as situation assessment based on DF. It is also important to understand that while these processes are interdependent, they are not connected by closed-form mathematics. Thus, evaluation of detection fusion performance does not lead to predictable tracking performance based on crisp mathematical interdependencies between the processes. It is also important to realize that conducting each level of analysis and evaluation requires either that amplifying test data be available to establish or estimate the applicable contextual condition, or that the necessary contextual information be supplied by the evaluation tool, such as a simulation model of some kind.

The complexity of this adjustment is made more difficult by the multi-platform scenarios that AFFTC will likely have to deal with. These factors generally make the performance of the system, and its effectiveness, *conditional* on various factors, which could be labeled "internal" factors and "external" factors. This terminology means that there are factors that can be controlled during the experiment by either the Test Analyst or the System Under Test (the "composite" test article)—these we label as "internal"—and those that are not controllable during the test but are set as part of the test plan—these we label as "external".

6.1.1 Data Fusion and Resource Management Processing Requirements

It is envisioned that there will be a progression of T&E activities at AFFTC that will in turn lead to a progression of DF and RM functionality of the type listed below (again, this is a draft, example list):

1. Onboard DF/RM for single platform
2. Onboard + Offboard DF/RM, for single and few platforms
3. Onboard + Offboard DF/RM, many platforms, highly controlled platform management
4. Onboard + Offboard DF/RM, many platforms, loosely controlled platform management; semi-autonomous
5. Onboard + Offboard DF/RM, many platforms, platform autonomy

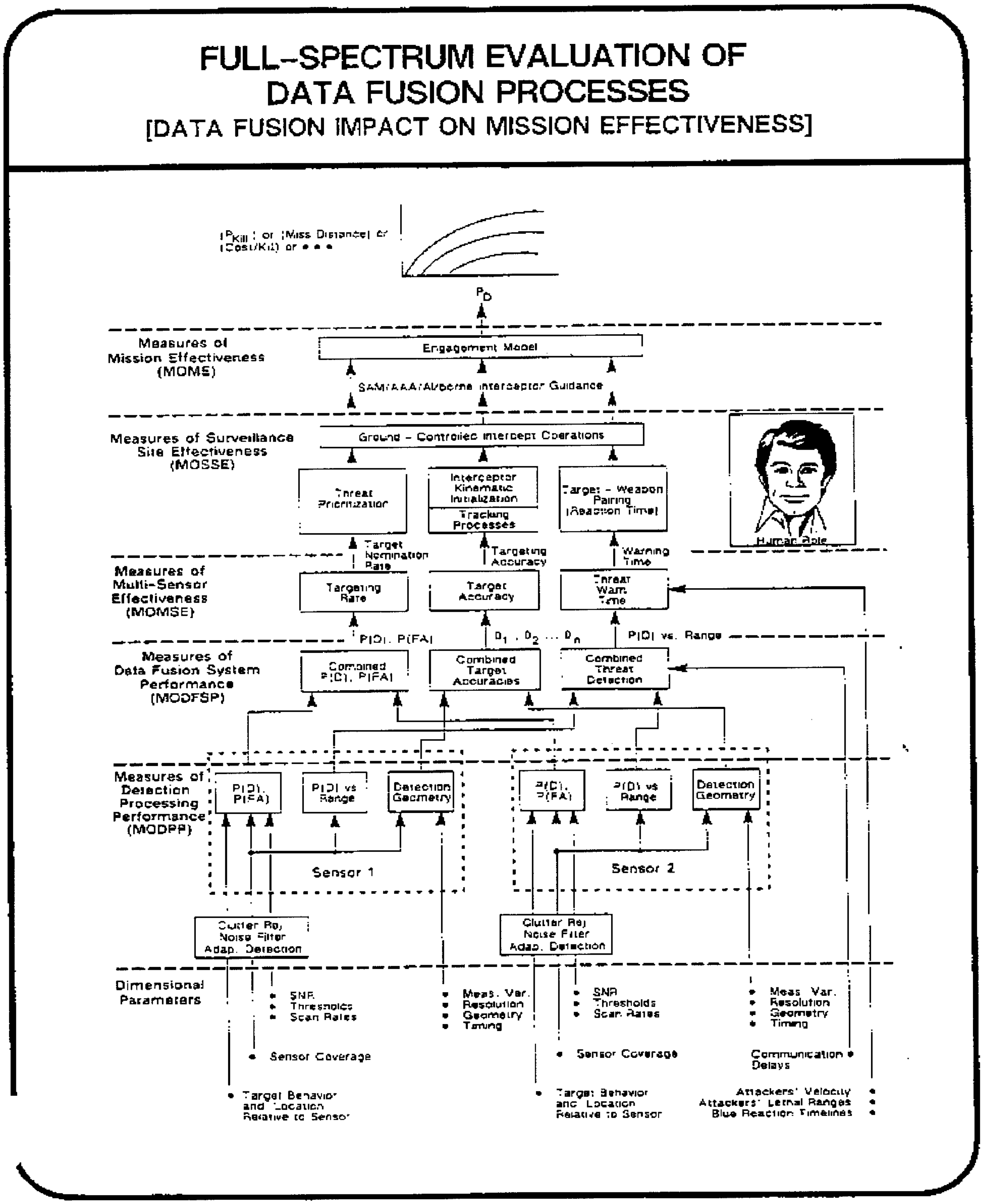

Figure 6.2. Concept of Layered Fusion Processes and Layered Evaluation Measures

Note too that this is just a "Blue" DF/RM list; how hostile, Red IADS DF/RM capabilities will grow over time is unknown, and for the intelligence community to define. White or Range-based DF/RM will also have to be re-engineered to exploit these techniques for the acquisition of improved range data. It can be seen from this list that there is a need to establish the Framework we are striving for, in that future test programs will increase the dimensionality and combinatoric complexity of overall test functions and activities. The idea of this Framework, as has been said previously, is to allow cost-effective growth at AFFTC to these higher levels of T&E capability. The "Onboard + Offboard" DF capability can be seen to grow toward a *distributed DF/RM process* almost immediately (as soon as there are a "few" platforms); we highlight this point as it implies growth in overall DF/RM processing complexity. Table 6.1 gives a *representative overview* of how the characteristics of both DF and RM change as the test scenarios change over time. By and large, the trend is toward test articles with ever-more intelligence and autonomy. This means that AFFTC will be evaluating not only the underlying DF and RM processes but the intelligence involved with: platform behavior, resource management (e.g. sensors and weapons), and the logics involved with the way in which information is shared among platforms, a crucial aspect of distributed DF.

In reexamining the DF/RM requirements, it is important that AFFTC carefully consider the scope and nature of the "Resources" that will be managed as part of the overall T&E domain, in essence as part of the overall test article concept. The extent of such resources can range over a fairly broad scale but it is thought that for AFFTC purposes these resources will encompass: platform and offboard sensors, comm. and data link system parameters as well as link or channel parameters, various aspects of the Information-Sharing Strategy (ISS), and the full gamut of countermeasures. Thus, the range of resources is really focused on those that influence the information quality and information flow; with an eye toward 'excluding' as noted above, this preliminary approach excludes most all *physical resources* such as weapon systems or platform management, etc.

6.1.2 Scenario Environmental Stimulus Requirements

Edwards AFB is a real, physical place on the Earth; it has therefore a limited spectrum of terrain variation, a limited extent of weather variation, and limited resources in terms of test support components and devices beyond the test article systems. While it may be able to simulate certain terrain effects (e.g. radar side lobe clutter from forest effects) through clever signal processing techniques, by and large the physical reality of the range will impose limits on what can feasibly be done in this regard. Limitations in the availability of various FME equipments and/or other components will also impose constraints on what equipment laydowns the range will be able to provide. Again, it is possible to provide the "presence" of certain equipments in a virtual way but this too involves a cost of development if the virtual device cannot be acquired somehow, as well as T&E (validation) costs and also deployment cost to install the device in the AFFTC Framework. In any case, these requirements also need formalization that will lead to component-level design for this function.

It would seem that a fundamental requirement for AFFTC with regard to providing not only mission-context related flexibility (representation of multiple platforms, support systems, weapon systems, etc) but also even simulated clutter or other environmental effects (e.g. terrain masking) that are unnatural to the California desert, is a capability for Distributed Interactive Simulation (DIS). In light of the

Table 6.1. Representative Overview of DF and RM Characteristics for Changing Test Scenarios

Phase of DF/RM Growth	General characteristics of DF	General Characteristics of RM	DF Level 1	DF Level 2	DF Level 3	DF Level 4	Network and Communications / Data linking	Information-Sharing Strategy
1	Centralized Msmt Fusion	Platform mgmt of organic resources (usually labeled "sensor management")	Msmt Fusion for Tkg; Feature fusion for ID	I&W	Missile launch; AAA firing	Organic sensor mgmt	Very limited	None
2	Limited Distributed Fusion	Well-defined resource mgmt protocols	Msmt and Track Fusion for Tkg; Feature and Decision Fusion for ID	I&W	Missile launch; AAA firing	Organic sensor mgmt and Resource sharing	Limited; some National Asset data links	Pre-mission briefed
3	Structured Distributed Fusion	Well-defined resource mgmt protocols with limited autonomy	Structured Distributed Tracking and ID Fusion	IADS Operational Phases	Distributed Threat Detection	Organic sensor mgmt and Resource sharing; Highly managed Platform Behaviors	"Regular" networks (Hierarchical, Tree, etc)	Predefined Push/Pull
4	Distributed Fusion with Intelligent ISS	Semi-autonomous; Teamwork-Based	Distributed Tracking and ID Fusion per ISS	IADS Operational Phases	Distributed Threat Detection and Threat Response	Partially Predefined and Partially Teamwork-Based	Decentralized Network Topology	Partially predefined, partially Intelligent
5	Distributed Fusion with Intelligent ISS	Autonomous; Teamwork-Based	Distributed Tracking and ID Fusion per ISS	IADS Operational Phases	Distributed Threat Detection and Threat Response	Partially Predefined and Partially Teamwork-Based	Decentralized Network Topology	Fully Intelligent

significant potential cost-effectiveness payoff of DIS, and associated with the post-Cold War decline in defense budgets, the topic of DIS has become of significant interest to the US DoD. With modeling and simulation offering potentially major benefits in various ways to DoD research and development, on June 21, 1991 the Undersecretary of Defense for Acquisition established the Defense Modeling and Simulation Office (DMSO) to serve as the executive secretariat for the Executive Council on Modeling and Simulation (EXCIMS) and to provide a full-time focal point for information concerning DoD modeling and simulation (M&S) activities. Currently the DMSO promulgates M&S policy, initiatives, and guidance to promote cooperation among DoD components to maximize efficiency and effectiveness. DIS was one of the focal points of the DMSO, and DMSO studied the issue of providing a common communication and execution infrastructure to allow standardized, cost-effective DIS to be achieved. Among other things, this led to the "High Level Architecture (HLA)", which is a common architecture allowing for reuse and interoperability across both technically and functionally heterogeneous simulators and also simulators that are geographically separated. An individual simulation or set of simulations developed for one purpose can be applied to another application under the HLA concept of the "federation", which is a composable set of interacting simulations. The core element of HLA is the Run-Time Infrastructure or RTI. RTI is in effect a distributed operating system for the simulation-federation. It provides a set of services that support the various simulations in carrying out federation-to-federation interactions and also federation management support services.

Notionally, RTI allows for interaction not only among digitally-based simulators but also—and importantly for AFFTC—it allows real players, e.g. real aircraft to be part of a federation; this idea is shown in Figure 6.3 below:

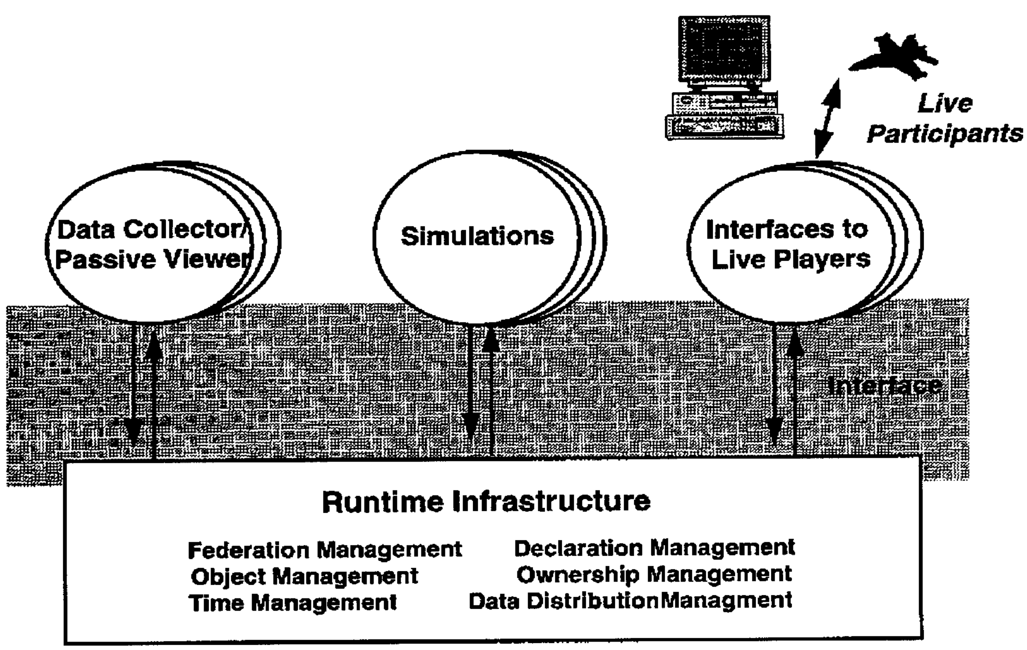

Figure 6.3. Notional Concept of the HLA Run-Time Infrastructure

Given our view that this type of capability will be needed at AFFTC as part of its future test support infrastructure, we at CMIF acquired the HLA RTI software (this is relatively straightforward to do), and have developed an easy-to-use GUI interface for possible future university-based research that would support further study of the future T&E support requirements for AFFTC. This GUI is shown in Figure 6.4.

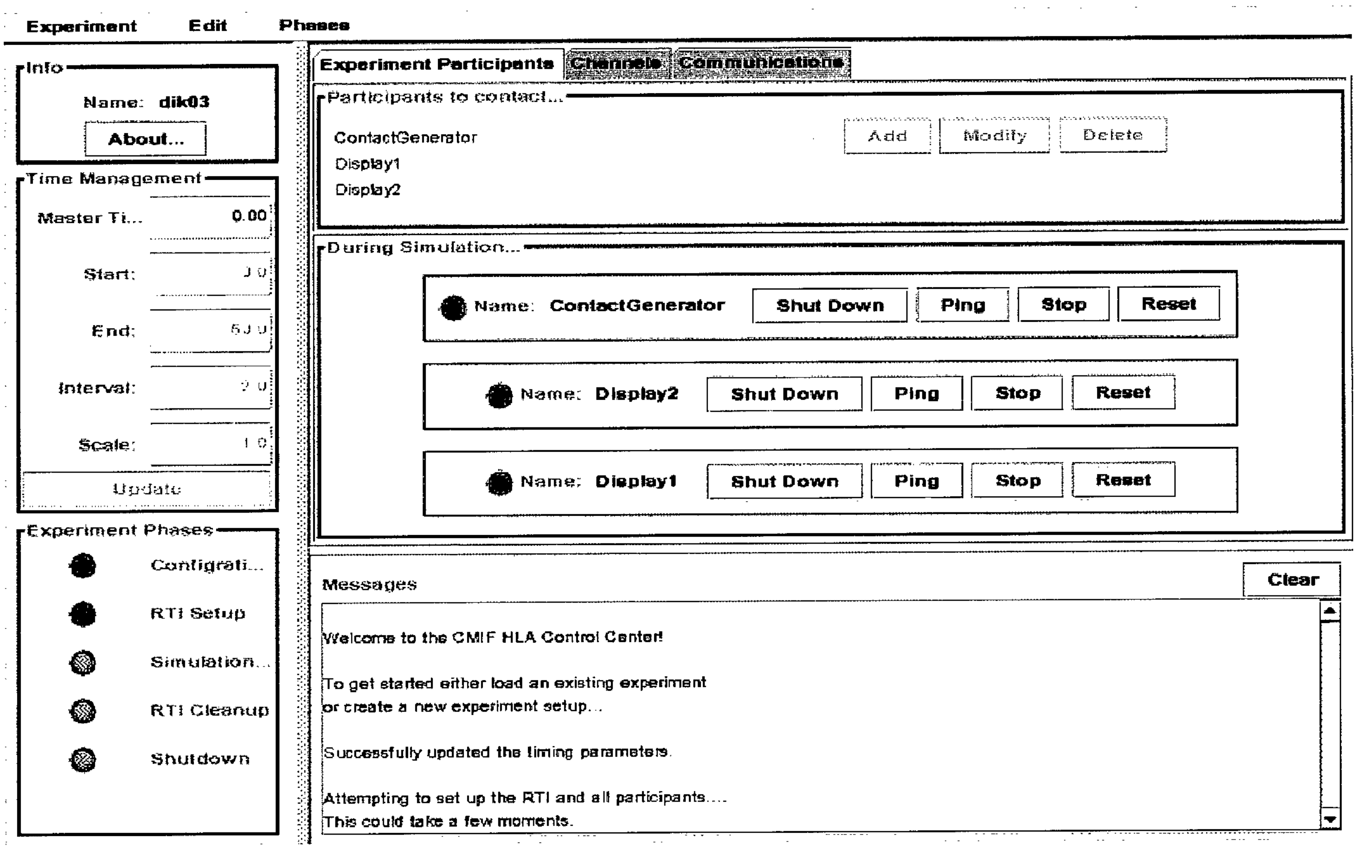

Figure 6.4. GUI Design for HLA/RTI-based Distributed Interactive Simulation Experiments in the CMIF Lab

The particular GUI instance shown in this figure is oriented to the establishment of a shared surveillance activity between two different sensor/fusion nodes, but the overall GUI design allows for a broad range of possible DIS-type multi-simulator operations. In designing this GUI, it was required to learn about the many different features allowed (and disallowed) in the HLA RTI environment; if in fact AFFTC will move toward HLA RTI for an eventual test support capability, staff from AFFTC will also have to learn the many options and features that RTI incorporates as part of the overall design process for that capability.

6.1.3 Performance Evaluation Requirements

This function is obviously one of the most important for AFFTC. The overarching requirement is for a PE process and system that has been validated, that is reliable, whose results can be developed in a timely fashion to satisfy user needs, and whose results can be properly couched in terms of the specifics of the test conditions under which the results were obtained.

Central to the Framework being discussed herein is the evaluation of DF and RM-related test events and data. Since it has been asserted that AFFTC can be expected to move toward OT&E from a DT&E legacy base, and that such testing will encompass a multiple platform, multiple sensor environment, it can be expected that the PE techniques necessary to evaluate DF and RM functions (i.e. all Red, White, and Blue variants of these functions) will have to be appropriate for the multi-object or multi-target case. If this assertion is correct, then the PE function becomes considerably more complex, because the traditional analysis of comparing test results to "truth" becomes more complex. In essence, the PE approach must be developed from the point of view of defining another DF process, i.e. a process developed from the same methodological approach employed for defining DF/RM processes in general. Although some thought has been given to the issues involved in defining such a PE approach (e.g. [Drummond and Fridling, 92]), defining a holistic, consistent approach to PE under these circumstances will require research into how to best define such a methodology. As can be seen by companion publications deriving from this project, we at CMIF have begun a research initiative in this direction. This topic was discussed at the June 5, 2000 AFOSR/AFFTC Review held at EAFB.

Definition of this PE process must also take a position with regard to the *temporal aspects* of analysis and evaluation. This means defining an approach to a dynamic technique to associate estimates produced by the DF/RM processes-under-test with either "known" truth (if based in a digital simulation) or "estimated truth" derived from the best calculations feasible from all available range data; i.e. fused range-data-based estimates (i.e. from White Fusion). Such comparisons could be made in accordance with sensor sampling rates or on a periodic basis or yet some other strategy; the point is that an approach must be defined on some rational basis. In addition, and in congruence with the temporal basis of analysis, an approach to MOP calculation must also be developed that will typically incorporate MOP's that are calculated for each time and then some type of approach to computation of cumulative MOP's.

Figure 6.5 below attempts to convey these ideas. It shows PE functional operations at two different times, with the scenario data (real or simulated, as noted) driving the "truth" states, from which are derived the (noisy) sensor observations. These observations feed the DF and RM processes-under-test, which, for Level 1 processing, generate "CTP Tracks", CTP meaning Consistent Tactical Picture, in that the DF and RM processes are considered as central to the establishment of a CTP. The truth tracks and the CTP track estimates are sent to the PE Node where both Current MOP's and Cumulative MOP's are computed. This process then continues at the next time epoch, as noted.

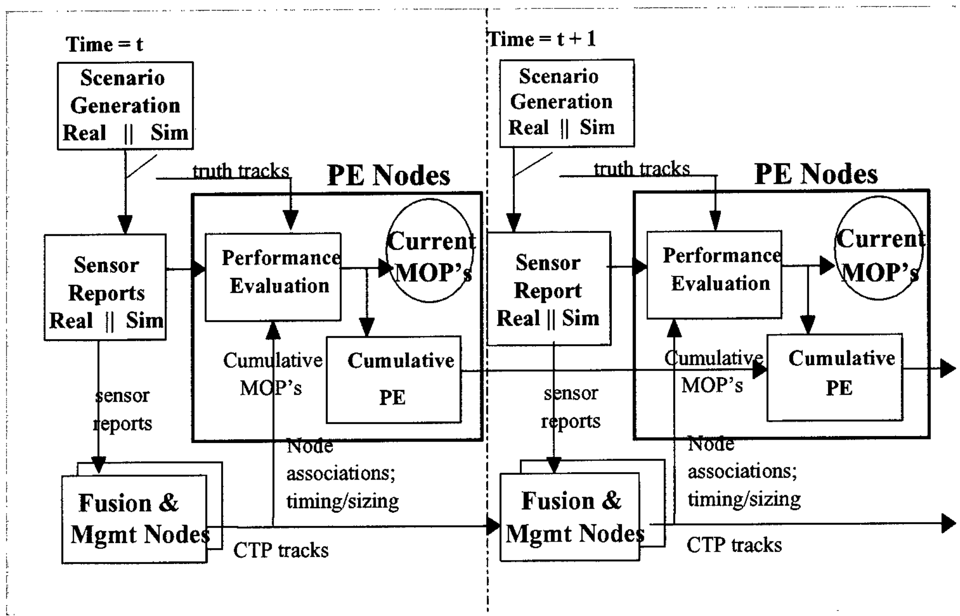

Figure 6.5. Depiction of PE Temporal Processing

6.1.4 Analyst I/O Requirements

Some type of test analyst position is envisioned in the BB Framework architecture. This person would be monitoring the data acquisition process as well as the evolving runtime or test-time results. This person would coordinate with the people performing the Test Management function if runtime adjustments were necessary to preserve the integrity of the acquired test data and the test results. Important to the execution of these functions is the need for a well-integrated analyst display system that conveys the status and operations of data acquisition and of MOP calculations as well as other PE runtime functions.

6.1.5 Test Management and Support Requirements

This important function, depending on how AFFTC will elect to develop its final BB Framework architecture, could be the function that oversees the configuration control of the Framework itself. It is clear that the Framework will need to evolve as future test requirements evolve, but it is very important that the Framework be kept under configuration control. It is envisioned that a Configuration Control Board would be set up for this purpose. Another crucial aspect of this function is control of applicable Data Bases necessary for a wide variety of test operations as well as PE processing. This function also supplies and controls range data linking and communication functions. If HLA technology and techniques are employed as the backbone data linking and process linking strategy for future test programs, then this function would oversee and control the HLA Run Time Infrastructure test configuration process and the control of inter-operating simulation processes as well.

6.1.6 Application Procedure Requirements for the BB Framework

We include these requirements since certain definitions of an "architecture" (e.g. IEEE) note that the definition of an architecture should include procedures for how to use or apply the architecture. The overarching requirement for the application of the BB Framework is that a methodology should be developed that minimizes the complexity and workload associated with using the Framework in any given test program. This hints at a requirement for automated support to this function. The basic task here is to "map" or transform from the concept of the test scenario to an allocation across real and virtual test support and test article elements, and then a mapping of these components to Framework components. This is in keeping with the very basic and underlying purpose of the Framework which is to provide a mechanism and structure with which to consistently envision and optimally design each test program involving DF and RM functions.

6.1.7 BB Component Design and Test Progression

The complexities and issues discussed above are, unfortunately, layered across a typical cycle of testing that, even for a single test program, ranges from simulation to flight test. The issues are driven not by the way the test is formulated or enabled, i.e. via simulation or hybrid or full-scale experiments, but by the underlying nature of the test scenario which leads to a staged or phased progression of tests using these various ways of representing the test conditions. The tradeoff is partially with respect to fidelity of representation versus validity of test results and extensibility of test results. All of this is of course intertwined with test costs. AFFTC needs, as part of defining the Framework, to establish a standard for this test progression process that would hopefully serve the needs of future test programs that involve DF/RM functions in some way or other. The following discussion, as for all parts of this document, is a draft version of such a test progression process.

Testing typically involves a progression starting anywhere from off-line, open-loop simple simulation scenarios to on-line, closed-loop, real-data and flight-testing, such as depicted in Table 6.2. This progression usually contains feedback cycles such that the real data evaluation results flow back to the simulation models to improve the validity of those models for subsequent experiments. This feedback process also aids in defining requirements for, and points to focus areas for further real data collection. The use of simulation or real data and the corresponding cycle for Aerospace Vehicle (AV) performance evaluation and experiments are defined in the AV Test Plan. The projected AV test spirals are also defined in the Test Plan, such as described in Table 6.3. Then the Test system requirements for each spiral are derived from the test requirements for each spiral to include the increasingly realistic experiment operating conditions.

Table 6.2. Representative Test Progression Process

Experiment Alternative Hierarchy	Off-Line Analysis Open Loop	Off-Line Analysis Closed Loop	Real Time Open Loop Experiment	Real Time Closed Loop Experiment	Flight Test
Test Environment	Lab	Lab	Mock Cockpits/ Dome Sim	Mock Cockpits/ Dome Sim	Air Vehicles
Simulation Drivers	Engineering Scenarios & Sensor Models	Engineering Scenarios & Sensor and Response Models	Human in the Loop and On-line Scenarios & Sensor Models	Human in the Loop and On-line Scenarios &Sensor/Respons e Models	Inflight Mission Training
Real Data Drivers	Archived Sensor & Link Data	Archived Sensor & Link Data	Real-Time Sensor Data Driven Fusion	Real-Time Sensor & Communications Data	Air Vehicle HW

Table 6.3. AV Testing Spirals

AV Test Phase/ Description	Spiral 1	Spiral 2	Spiral 3	Spiral 4
Operations Environment				
Experiment Type				
AV Nodes				
Sensors/ Sources				
Data Characteristics				
Fusion Capability				
Resource Mgmt Required Capability				
System Modes				
Performance Evaluation				
User Interface & Reporting				

6.2 AFFTC Test Framework Component Design Optimization

A very preliminary, top-level component design is shown in Figure 6.6. This figure just shows the notional interfaces between the functions allocated to the BB Framework. The means of functional connectivity will vary between some network structure and some onboard bus for specific platforms or integrated systems. The interfaces to the external functions, not an integral art of the BB Framework, can be enabled in various ways or stubbed out as non-functional depending on specific test requirements.

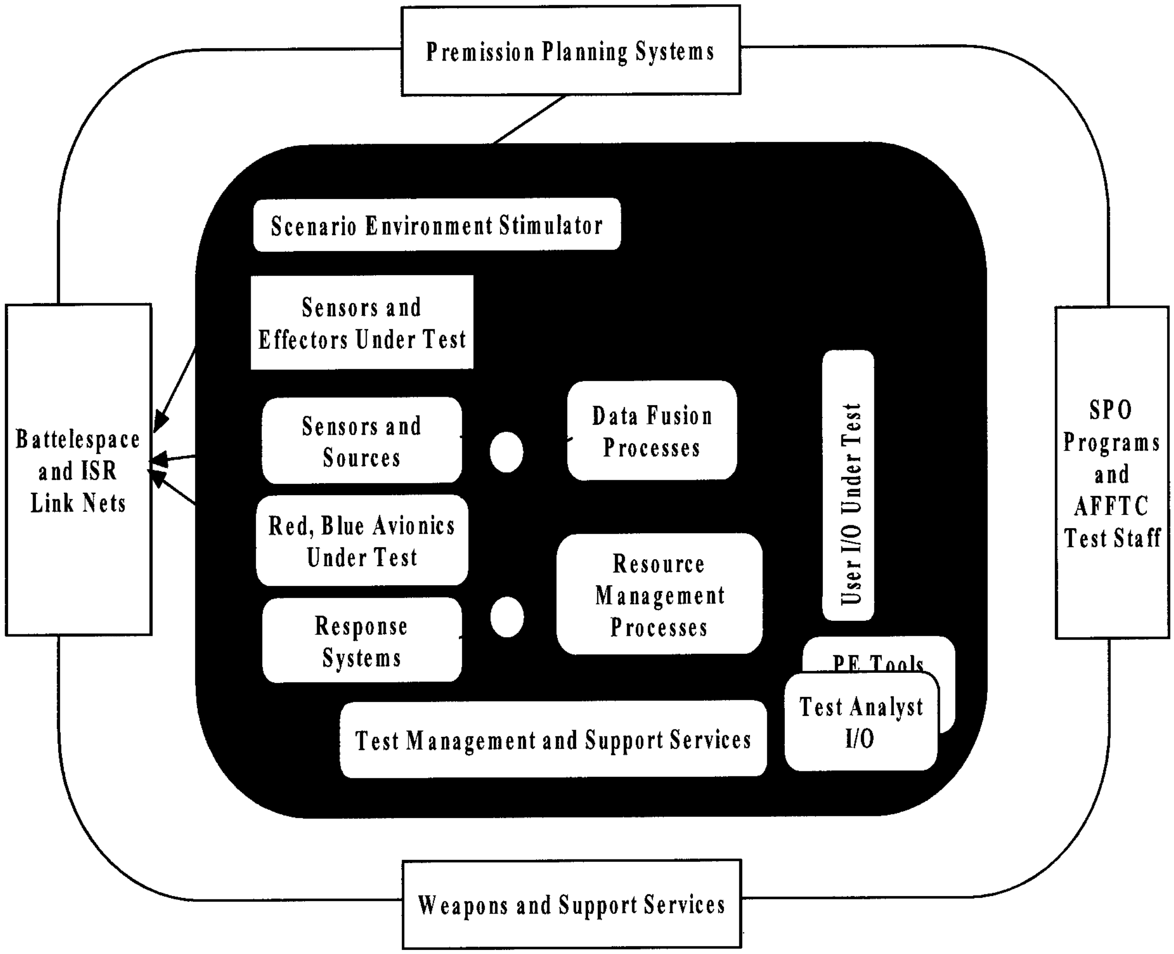

Figure 6.6. Notional, General Depiction of BB Framework Components

Figure 6.7 is an expansion of Figure 6.6, now showing how the BB component design would seem to provide the basic building blocks to represent various test conditions. The 4-component sets {Sensor & Sources (S&S) + Data Fusion Processes (DFP) + Resource Management Processes (RMP) and Response Systems (RS)} are shown as "nodes" in the BB Framework for this typical test condition depicting the platform and other integrated-unit information processing operations. The support functions are shown around the periphery. Note too that White Data Fusion, depicting range resources and associated DF processing, are shown in the same way.

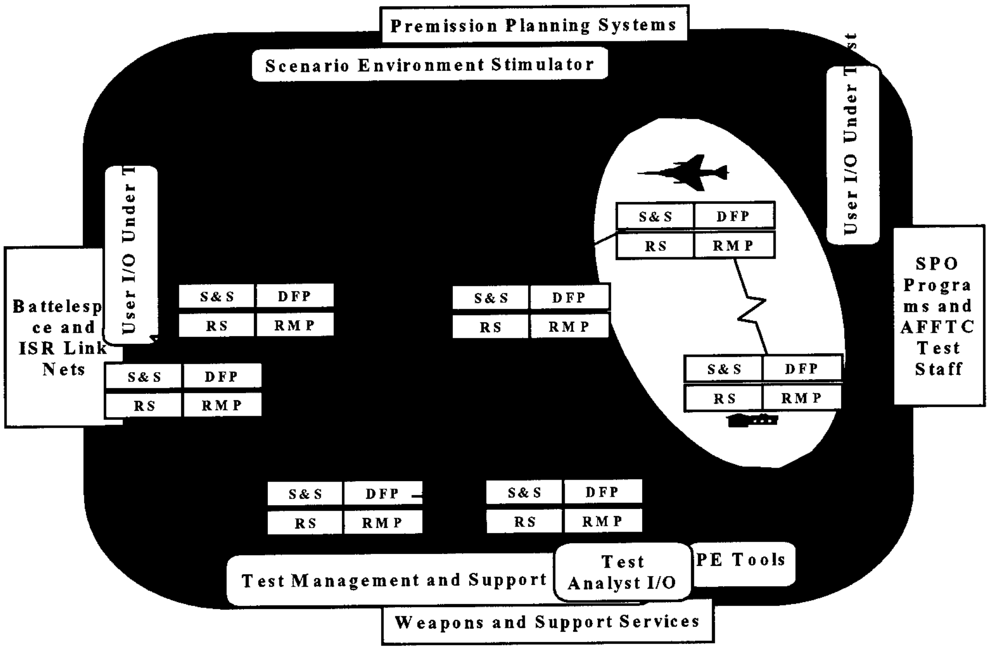

Figure 6.7. Distributed Test Environment Using BB Design Components

Finally, Figure 6.8 shows the "physical" analogy to these ideas, in terms of actual and virtual range test platforms and components. In practice, it is more likely that the real-world components would be defined first, with the abstraction to the process and BB element levels following next. It is these types of conceptual iterations that must be carried out to evolve a solid definition and characterization of the Framework we have been discussing herein.

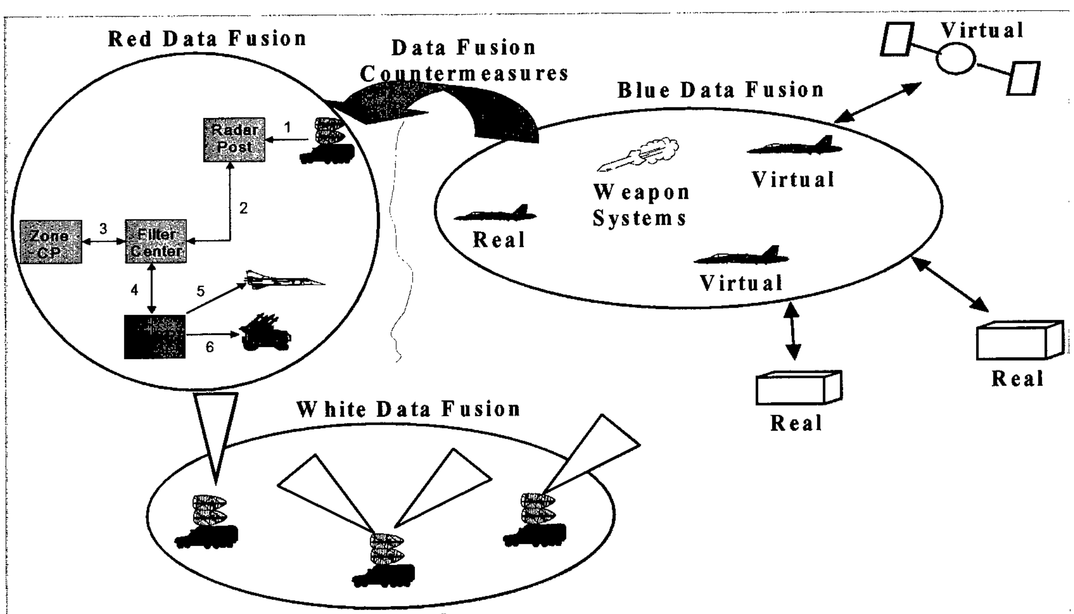

Figure 6.8. Physical Analog of Distributed Data Fusion Test Environment

6.2.1 PE Component Design

The PE component design must address the PE requirements described above. In essence, as noted in Section 6.1.3, the PE process is a reduced-form of a fusion process. That is, it will have a variation of a typical fusion node's processing operations. If we recall the "normal" Fusion Node, it looks like:

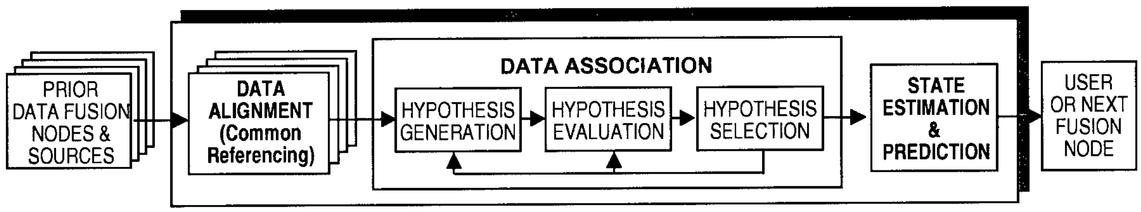

Figure 6.9. Typical Fusion Processing Node

In our first-cut design approach herein, we define the "PE Node" design to have a similar structure, as shown in Figure 6.10 below: There is some degree of Data Preparation that we would envision as typically necessary, similar to the "Data Alignment" function for typical DF processing. There is however an equally-serious and important Data Association process design that must be formulated, since the strategy by which truth information is compared to "information-under-test" is the heart of the PE process, and will govern the way in which MOP's are computed and what their values will be. Thus, once the PE Data Association process is complete, in a way similar to the same step in conventional DF, there will be assignments of test data to truth data, which then forms the basis for the MOP calculations (MOP State Estimation in the figure); we label these as estimates since the test data-to-truth data assignments will generally be imperfect in some way and to some degree.

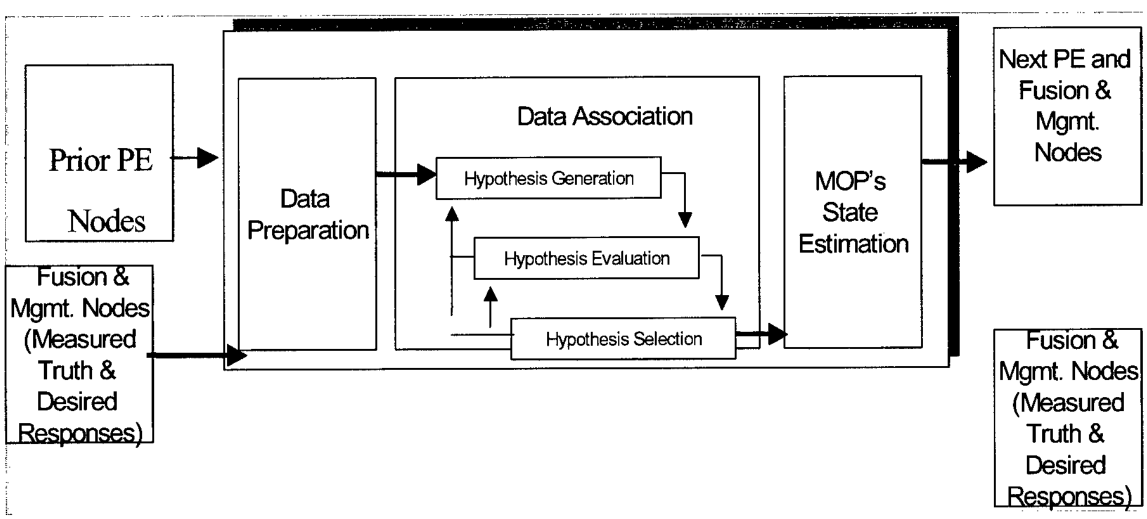

Figure 6.10. Generalized Component Design for PE Node

6.3 AFFTC Test Framework Component Design Evaluation

6.3.1 A Viewpoint on Design Evaluation

An attempt has been made to describe the Framework and how it should be defined and developed by AFFTC to be consistent with the principles of Object-Oriented Design (OOD) as used most typically for the design of software systems. A fundamental point of course is that object-oriented software is all about objects. In turn, an object is a "black box" which receives and sends messages. A core design aspect is encapsulation or an approach that hides the object's functionality from the message-generation and passing processes. An object is defined via its class, which determines everything about an object. Objects are individual instances of a class. The term "method" is also used on OOD; a method is simply the action that a message carries out. It is the particular instance of the object's function which gets executed when a message of a particular structure is sent to a particular object. Further, there is the notion of inheritance in which, when defining a new class, that class inherits the behavior of the parent class. In defining the object classes and the overall system architecture, it is important to define a level of abstraction that will be applied in performing the design; this is another design choice. For example, in all of the above we are at a component level of abstraction, selected here largely on the basis that it is adequate to communicate the ideas of the Framework concept. As a consequence of choosing an approach having analogies to OOD, one reasonable approach to evaluation derives from representative methods for evaluating OOD's. Notice that these methods can be used both in the initial construction of the Framework design but also to maintain an evolving, changing design.

6.3.2 Design Evaluation Metrics

No matter how well-done the derivation of the Framework requirements has been done, those requirements will not likely lead to a singular point design for the Framework. That is, the design process is not a deterministic, closed-form process. If this is true, then some basis for comparing alternative Framework designs should be established. This evaluation procedure will evolve from requirements as well. As a representative approach, and following the argument above, we describe here typical metrics used in evaluating OOD's; those discussed here were taken, almost literally, from a website at: http://ivs.cs.uni-magdeburg.de/sw-eng/us/experiments/chik/; see also [Chidamber, 93]. In this referenced work, the following metrics are explained in detail:

- Weighted Methods Per Class (WMC)
- Depth of Inherence Tree (DIT)
- Number of Children (NOC)
- Coupling between object classes (CBO)
- Response For a Class (RFC)
- Lack of Cohesion in Methods (LCOM)

In what follows, we discuss a couple of these metrics, again deriving the remarks almost literally from the web reference.

Weighted Methods Per Class (WMC)
Consider a Class C1 with methods M1,...,Mn that are defined in the class. Let c1,...,cn be the

complexity of the methods. Then:

$$WMC = \Sigma\ c_i\ ,\ i = 1, n$$

Why is this particular metric used? It is because the number of methods and their complexity are a measure of how much time and effort are required to develop and maintain the class. The greater the number of methods in a class the greater the potential impact on children because they inherit all methods defined in the class. Classes with large numbers of methods are likely to be more application specific, limiting the possibility of reuse.

Depth of Inherence Tree (DIT)
Depth of inheritance of the class is the DIT metric for the class. In cases involving multiple inheritance, the DIT will be the maximum length from the node to the root of the tree. It is a measure of how many ancestor classes can potentially affect this class.

This metric is employed to evaluate OOD's because the deeper a class is in the hierarchy, the greater is the effort to predict its behavior, because of the likely greater number of inherited methods. Deeper trees constitute larger design complexity, because more methods and classes are taken into consideration. The potential reuse of inherited methods increases according to the depth of the considered class.

Additional metrics are discussed in Appendix A.

6.3.3 Framework Design Control

While we have discussed the notion of design evaluation and design evolution, the establishment of the initial Framework design and especially the evolution of that design should be placed under some type of control process, just as we have suggested for the overall Framework as well. Continuing the idea of similarity of this overall Framework process to a software architecture, a Configuration Control Board would be an excellent way to properly maintain the Framework design as changes are made to accommodate evolving T&E requirements.

6.4 Framework Detailed Design and Framework Design Patterns—Phase 3

Once the component-level design is complete, the cycles depicted in the DNN design process of Figure 3.3 continue to the detailed level, which could be called a 'nodal' level. Here, starting again from requirements definition for these nodal partitions, a design tradeoff and definition process is begun. It would be expected that many existing algorithms and techniques used for conventional (application-oriented) DF and RM process designs would be defined as reusable nodal elements of the Framework. Examples might be the wide variety of assignment algorithms, object classification algorithms, tracking algorithms, etc that exist today and are generally available as open-source, shareable knowledge and software codes.

We emphasize that throughout this Framework document when we speak of fusion processes we mean the full breadth of Blue-Red-White fusion processes, not just Blue processes. In part to emphasize this point, the role of Red data fusion in typical hostile IADS is discussed in Appendix B, as are some representative techniques for data fusion countermeasures. What is important at this stage of the Framework definition process is to develop a cost-effective approach to representing the type environment shown above for the spectrum of test programs AFFTC expects to encounter over the near to far-term future. In terms of detailed design, at least one key issue is that of the level of fidelity at which each of these components will be provided. It may be possible to determine that certain elements can be provided best at a fixed level of fidelity (this is an important decision), whereas others will have to be provided at varying levels of fidelity. What must be done of course is to study both the hostile IADS and friendly test-article structures expected in future test programs and map them into a notional depiction as above and also into the data fusion "levels" of the standard data fusion process. The criticality of each such component to the performance and effectiveness assessments resulting from each test must also be determined at least notionally if not specifically. Further, the role of White data fusion, or the data fusion processing to be provided within the range sensor and processing system, must also be determined; this is a matter that is in AFFTC's hands and has to do with the question of how good the "truth" of any given test scenario needs to be determined. A major factor in all the analyses described here is that of the real-time requirement for any given function. For example, if there is active sensor management within a given test case, it is likely that such functionality will have to be provided at real-time, since post-test simulation of such dynamically-acquired observations would probably not be possible or provide realistic results. This doesn't mean that the function must be provided at high-fidelity but that it must be provided at real-time; i.e., real-time and high-fidelity are not necessarily correlated.

According to Kurotsuchi (see Design Patterns Tutorial at http://www.csc.calpoly.edu/~dbutler/tutorials/winter96/patterns/), the origin of design patterns lies in work done by an architect named Christopher Alexander during the late 1970s. He began by writing two books, A Pattern Language (see [Alex77]) and A Timeless Way of Building (see [Alex79]) which, in addition to giving examples, described his rationale for documenting patterns. Software patterns first became popular with the wide acceptance of the book Design Patterns: Elements of Reusable Object-Oriented Software by Erich Gamma et al; see [Gamma, 94]. Due to the overwhelming acceptance of this book, much of the initial patterns focus in the software community has been on design patterns. The patterns in the book are object-oriented design patterns. There are many other kinds of software patterns besides design patterns. In the software world, a pattern is generally defined as: a named nugget of insight that conveys the essence of a proven solution to a recurring problem within a certain context amidst competing concerns. A more expanded characterization is as follows:

A *design pattern* names, abstracts, and identifies the key aspects of a common design structure that makes it useful for creating a reusable object-oriented design. The design pattern identifies the participating classes and their instances, their roles and collaborations, and the distribution of responsibilities. Each design pattern focuses on a particular object-oriented design problem or issue. It describes when it applies, whether or not in can be applied in view of other design constraints, and the consequences and trade-offs of its use. Since we must eventually implement our designs, a design pattern also provides sample code to illustrate an implementation.

Although design patterns describe object-oriented designs, they are based on practical solutions that have been implemented in mainstream object-oriented programming languages. The goal of patterns within the software community is to create a body of literature to help software developers resolve recurring problems encountered throughout all of software development. Patterns help create a shared language for communicating insight and experience about these problems and their solutions. Formally codifying these solutions and their relationships allows the successful capture of the knowledge that defines our understanding of good architectures that meet the needs of users.

We introduce and suggest the use of such design-pattern ideas in the context of the design of the Framework and its detailed components and nodes because of the overall cost-effectiveness and minimal lifecycle cost implications of their use. It is presumed that the Framework will evolve over time but its careful initial definition and methodology of construction will set the tone for its overall utility and ease of modification for as yet unforeseen new aerospace vehicle concepts and concepts of employment.

References

[Alex, 77] C. Alexander, S. Ishikawa, M. Silverstein, M. Jacobson, I. Fiksdahl-King and S. Angel, A Pattern Language (New York: Oxford University Press), 1977.

[Alex 79] C. Alexander, The Timeless Way of Building, Oxford University Press, 1979.

[Chidamber, 93] Chidamber, S.R.; Kemerer, C.F.: A Metrics Suite for Object Oriented Design. CISR Working Paper No. 249 (Cambridge: M.I.T), February 1993.

[Gamma, 94] Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides, Design Patterns: Elements of Reusable Object-Oriented Software, Addison Wesley, October 1994.

[Drummond and Fridling, 92] Drummond, O.E. and Fridling, B.E.: Ambiguities in Evaluating Performance of Multiple Target Tracking Algorithms, SPIE Vol. 1698, Signal and Data Processing of Small Targets, 1992.

7.0 Using The Framework

AFFTC must of course evaluate the ideas being set forth in this document, and even if it is judged that some type of Framework will be developed, an AFFTC approach to using the Framework must also be established. Our opinion is that one of the first things to be done is to decide on an organizational approach to this question; issues of oversight, control responsibility, and change authority, and also the very important aspect of rigor in application must be decided first. Further, there will be decisions required on how, and which part of, the AFFTC staff will be trained on what the Framework is and how it will be used. That is, the first questions relate to inculcating the Framework notion and its use as a matter of organizational culture, and achieving "buy-in" by the operational staff.

This being said, it would seem that in the early implementation of the Framework that a dedicated team will be required to conduct the analyses of any given test program and carry out the mapping of that program's requirements to the Black Box components and related detailed nodal elements. As will likely happen in many real-world cases, there will be exceptions that will arise and these will have to be worked out on a case-by-case basis, requiring negotiation both internal to AFFTC and externally with a test program sponsor. It is exactly these exceptions, as well as advances in technology and platforms, etc, that will create the evolutionary path for the Framework. No doubt Framework configuration control will become an issue, and the procedures for decision-making with respect to changes will have to be defined within the Configuration Control Board.

At an appropriate point in time, training of the AFFTC will be carried out and the Framework process then becomes part of the overall test planning and execution process within the overall organization. There will still need to be a CCB and some type of central control, until the next revolutionary cycle in test concepts, planning, and execution occurs.

8.0 Summary & Recommendations

Concepts like the Framework described here are difficult to sell to management; they are among the concepts and procedures related to achieving long-term benefits via strategic redirection or modification. A typical managerial response is one that understands and possibly even applauds the idea but argues against implementation based on concerns for near-term pressures and issues. As mentioned above, inculcating the Framework way of thinking and in addition implementing it in some widespread way effectively involves a cultural sea-change in the way of doing business at AFFTC, and so will give pause to managers.

We believe the best next action to take is to conduct a Case Study; this too however will require a degree of buy-in by some subset of technical staff at AFFTC since their participation and guidance will be crucial to achieving a fair and equitable evaluation of the Framework's utility and implied cost-effectiveness. Done right, the Case Study will form a solid and quantitative basis for more serious judgments about next steps.

Appendix A. Object-Oriented Design Metrics

This Appendix is derived verbatim from website http://ivs.cc.uni-,agdeburg.de/sw-eng/us/experiments/chik/ as was used in describing some of the other metrics in Section 6.3. The additional metrics, not discussed in Section 6.3, are:

Number of Children (NOC)

Definition:

Number of children = number of immediate subclasses subordinated to a class in the class hierarchy. It is a measure of how many subclasses are going to inherit the methods of the parents class.

Consideration:

The reuse is in direct proportion with the number of children, since inheritance is a form of reuse. The existence of a class with a great number of children may mean a case of misuse of subclassing, because the probability of improper abstraction of the parents class is high. The NOC value gives an idea of the potential influence a class has on the design.

Coupling Between Object Classes (CBO)

Definition:

CBO for a class is a count of the number of other classes to which it is coupled.

Consideration:

Inter-class couples should be minimized as much as possible, because of reusability, maintenance and modularity. This measure is useful for determining the testing complexity.

Response for a Class (RFC)

Definition:

RFC = |RS| where RS is the response set for the class.

Consideration:

The greater the number of methods can be invoked in response to a message the greater is the complexity of class and thus the testing and debugging effort.

Lack of Cohesion in Methods (LCOM)

Definition:

Consider a class C1 with n methods M1,M2,...,Mn. Let {Ij}=set of instance variables used by method Mi. There are n such sets {I1},...,{In}. Let P = {(Ii,Ij) | Ii joined with Ij = 0} and Q = {(Ii,Ij) | Ii joined with Ij <> 0}. If all n sets {I1},...,{In} are 0 the let P = 0.

$$\text{LCOM} = \begin{cases} |P|-|Q|, & \text{if } |P|>|Q| \\ 0 & \text{otherwise} \end{cases}$$

Consideration:

Low cohesion of methods implies a large likelihood of errors during the development process, because of the increasing complexity. It can be measured whether a class should be split into subclasses. All in, all cohesiveness of methods within a class is desirable, since it promotes encapsulation.

Appendix B. Modern-Day Integrated Air Defense Systems (IADS) and the Role of Data Fusion

B.1 Basic Concepts of an IADS

An Integrated Air Defense System is the structure, equipment, personnel, procedures and weapons that are used to counter the enemy's airborne penetration of one's own claimed territory. While this section focuses on "modern-day" IADS, it should be recognized that there is a spectrum of capability and operational characteristics across any representative system deployed today. The equipment types run the gamut from the very old to modern systems, including older sensors to modern up-to-date sensors, older computer systems to modern up-to-date computer systems, and older communications systems to modern up-to-date communications. The levels of capability and training of IADS personnel also varies widely, as does the degree of adherence to declared procedures. The weapons systems of an IADS typically fall into three major categories:

AAA Guns: older point and shoot to modern radar controlled guns usually used for point defense

SAM systems:

- Short range used for point defense.
- Medium and Long range systems used for point and area defense

AI Systems: Both older and modern systems used for area and long range defense.

While an IADS can be employed for missile defense, our concerns in relation to AFFTC's T&E focus will be on Counter-Air operations, which are oriented to protect ground forces and critical assets from attack by enemy fixed- and rotary-wing aircraft and unmanned aerial vehicles (UAVs).

For the IADS under attack, the threat is not limited to just attack aircraft; the threat includes all aircraft, such as aerial surveillance platforms, unmanned aerial vehicles, cruise missiles, and satellites, when these systems are working in unison to execute coordinated attacks. This is exactly the type of future environment we are focused on in considering the evolving T&E needs for AFFTC.

Perhaps the most important aspect of an IADS to understand is its C2 system and procedures. The control of an IADS is relatively complex, involving:

- Weapon control procedures
- Coordination with adjacent AD units
- Coordination between service components
- Through shared knowledge of the enemy and friendly situation

Crucial as always to the effective employment of C2 procedures is the information flow that supports them. An IADS, requires the provision and exchange of essential real-time information, including:

- Air defense warnings that allow commanders to implement the appropriate active and passive air defense measures.
- Adequate track capacity within systems and the cross-telling of tracks using data processing systems.

- Both space-based and ground-based secure communications assets.

Execution of counter air operations requires a surveillance and reporting system capable of near-real-time production and dissemination of tracking data necessary for the effective engagement of targets.

B.2 Data Fusion in a Typical Threat IADS Structure (the Defensive Role)

A typical threat Integrated Air Defense System (IADS) structure is illustrated in Figure B.1. As shown in the figure typical structures are layered in a hierarchy. The number of

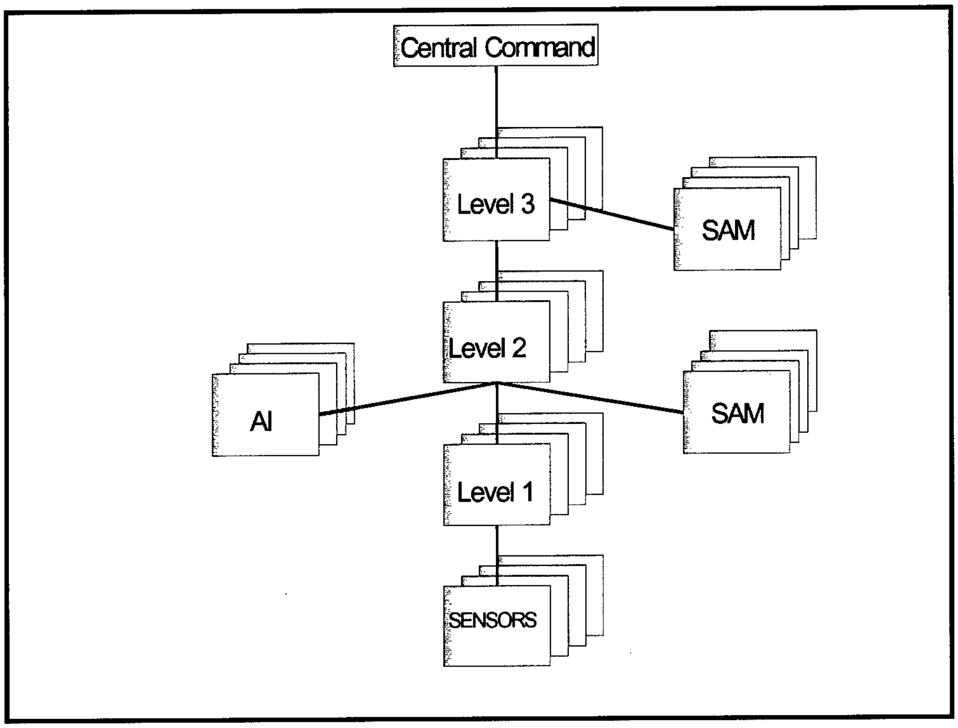

Figure B.1. Typical IADS Structure

levels vary significantly both between and within countries. At leach level of the IADS structure in Figure B.1 some form of data fusion processing takes place. At the lowest levels detections from the sensors are fused together to form tracks. Other sensors are then cued to provide IFF and height information. As this information flows up the chain more data is added and a complete air picture of the hostile and friendly situation is developed. The IADS uses this air picture to make engagement decisions on what targets to engage, what to engage these targets with, and when to engage these targets. Once decisions are made, the IADS weapons (Anti aircraft artillery (AAA), Surface to Air missile (SAMs) and Airborne Interceptors (AIs)) are commanded to complete the engagement actions. Functionally, these IADS processes can be segmented into three distinct areas as shown in Figure B.2. As shown in the figure, these areas are

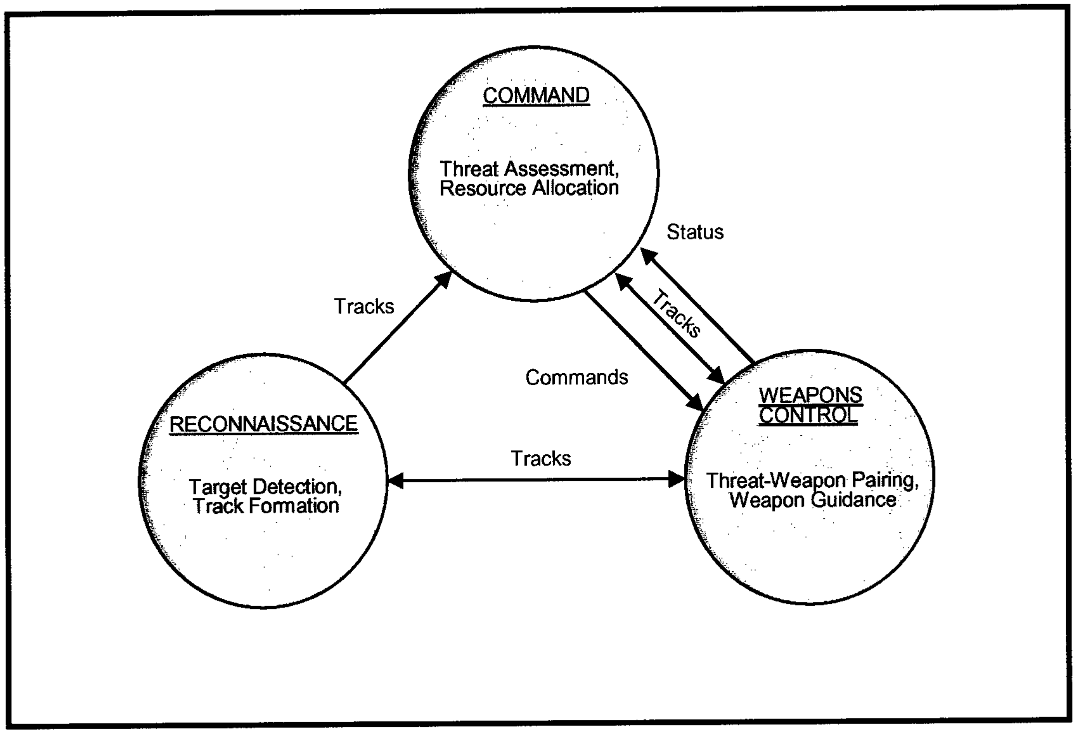

Figure B.2. Functional IADS Structure

Reconnaissance, Command and Weapons Control. Any given node in the IADS structure in Figure B.1 performs one or more of these functions. Data fusion is a critical component of all three functional areas. Note that these functions can be, and usually are, realized in a network-type structure, employing a set of nodes which each may be configured as a single-sensor or multi-sensor subsystem itself, usually communicating in a hierarchical comm.-system structure.

B.3 Overview of IADS Data Fusion Processing

Any threat IADS is generally based on the concept of aircraft tracks. A track is the accumulation the *estimated* information on an *estimated* aircraft at any given IADS node. Figure B.3 provides a typical list of data items that are fused to form an IADS track.

As new information is obtained, it is used to create or update the estimated aircraft track. Position estimates might be obtained from a 2D early warning sensor, height from a height finder sensor, IFF information from an IFF sensor, aircraft type from a visual observer etc. The IADS track represents the total of all fused information on each perceived aircraft. The track information is updated and refined by each successive IADS node. The accumulation of all IADS tracks forms the IADS air picture that is used to make engagement decisions. So in summary, the IADS fuses information into tracks, forms an air picture from the tracks and engages selected priority tracks. It is important to note that the track is the perceived or estimated information on an aircraft from the IADS perspective, not necessarily the true air picture.

- Time
- Track Number
- Friend/Foe
- 2D Position
- Altitude
- Velocity
- Heading
- Track quality
- Engagement status
- Position source
- Raid size
- Type
- Fuel (Friends only)

Figure B.3. Typical Track Data

The reconnaissance functional area is the major functional user of data fusion processing. In reconnaissance, the actual IADS tracks are formed; once the track is formed, data from all reconnaissance sources are fused to update the IADS track. If the data within a track is deemed sufficiently old, the track is dropped. When the track is deemed of sufficient quality it is subject to be transmitted to other IADS nodes. The basic tracking process can be viewed as consisting of three steps. These steps are illustrated in Figure B.4. As shown in the figure these steps are designated Primary processing, Secondary processing, and Tertiary processing. Each processing operation of a threat IADS will use different software and algorithms to implement these three processing steps.

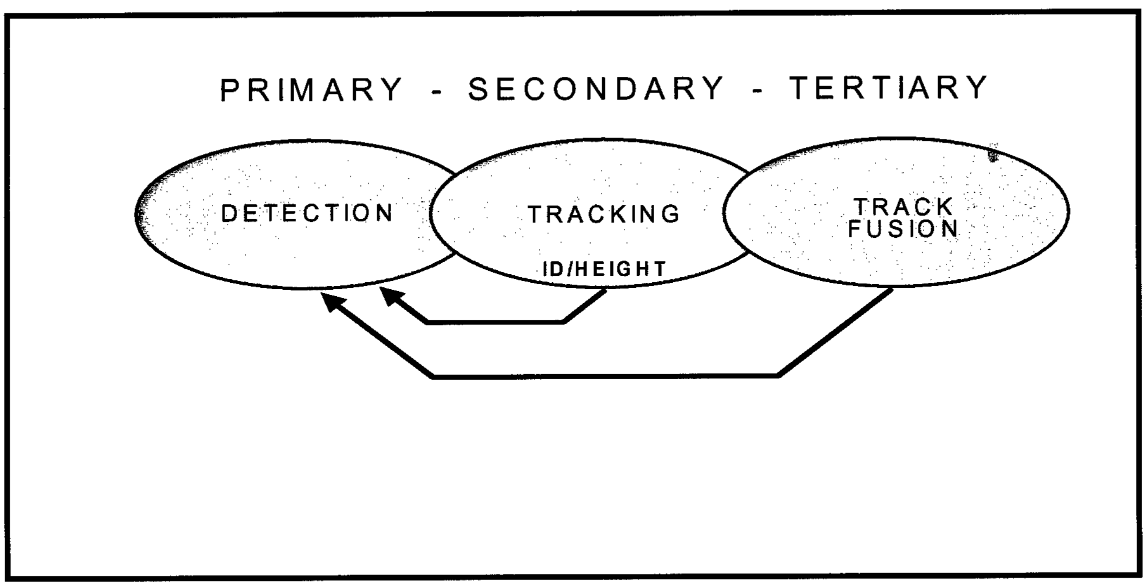

Figure B.4. Typical Reconnaissance Processing

The data fusion processes within the modern integrated IADS are software-based processes. These modern processes tend to be much more automated. It is not that the human is no longer involved, but that the role of the human is much different. In older systems the human actually performed the detection and tracking and fusion processes. In modern systems, the human monitors the systems and the software performs the actual work. The actual engagement decisions are formulated by the software and recommended to a human for confirmation. If everything fails, the human will indeed take over, but unlike before the human being is not trained to operate the process manually and the equipment is not easily operated in a manual mode.

The next few paragraphs will provide an overview of these basic processing steps. Regarding AFFTC's interests in data fusion and data fusion countermeasures, we will see later that knowledge of the actual algorithms and software employed at each level is essential for successful employment of countermeasure techniques to the data fusion component of an IADS.

Primary processing refers to the actual detection process at a sensor as illustrated in Figure B.5. During the detection process a contact is obtained, and its coordinates are digitized. This process can be either automatic through an extractor or manual through an operator. Virtually all-new sensor systems contain built-in extractors.

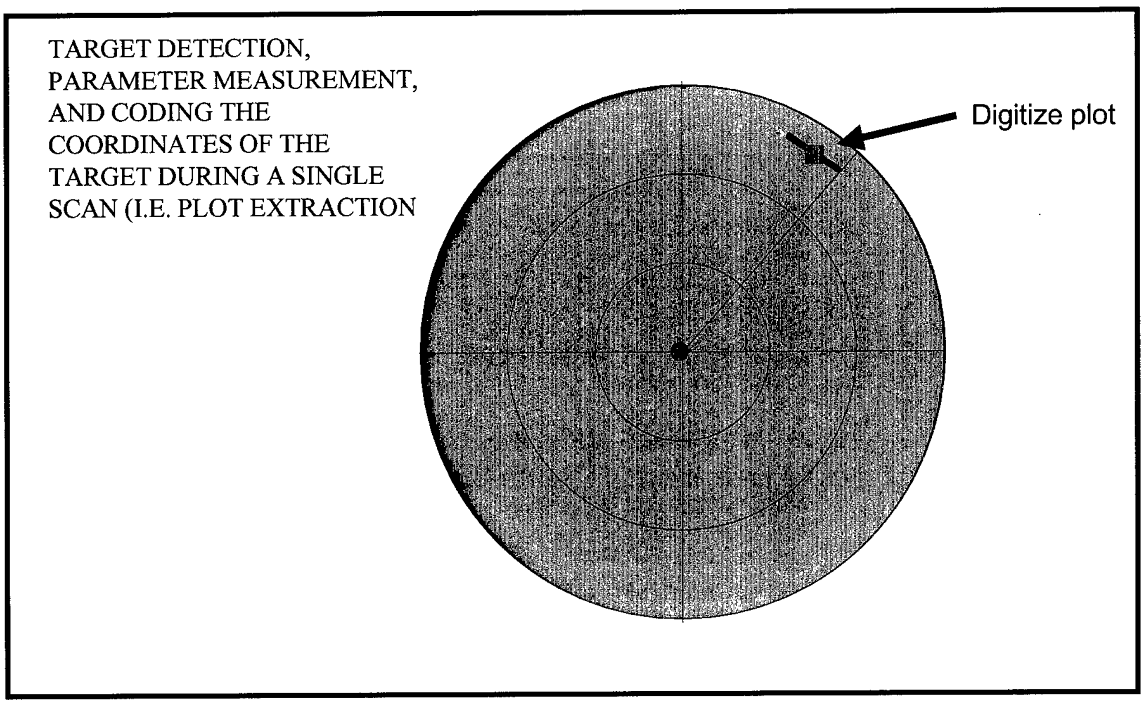

Figure B.5. Typical Reconnaissance Processing

Secondary processing refers to association processing of the primary-level contacts to create or maintain a track. This can be done either individually by sensor in what is called single sensor tracking or across sensors in what is termed (for IADS systems) "plot fusion" tracking.—otherwise called measurement fusion in the general data fusion community. Both types of tracking systems are found within modern IADS structures. However the single radar tracking approach dominates today, and will probably do so into the near future. The secondary processing component of modern IADS systems contains algorithmic logic to initiate, update, smooth, and drop tracks; it is probable that any modern system is taking advantage of all the different flavors of tracker techniques in these algorithms. This is all performed automatically with virtually no operator intervention in most systems built or deployed today. If plot fusion is employed, the final fused air picture results from this step, and the output of the secondary processing component drives the logic that determines when and where engagements will take place. Secondary Processing is illustrated in Figure B.6.

Tertiary processing refers to the combining of track data across track sources.—this is the fusion of local track state estimates, typically called "track fusion". Tertiary processing is illustrated in Figure B.7. IADS tend to incorporate very redundant structures with the same air objects being tracked by different sources. The tertiary processing functional area correlates the track data from these diverse sources and then combines it to form a best system estimate of all the variables within the track file.—for the case where tertiary processing combines track estimates (top figure), correlation of track estimates and subsequent track fusion is a very tricky process requiring awareness on the part of adversaries to the intricate technical details, otherwise this process will be corrupted. As recently as 1998, subtle mathematical details were uncovered in the "traditional" approach to track fusion employing a weighted covariance approach and involving the so-called "cross-covariance". Violation of certain constraint conditions by the nodes sending their track estimates to be fused would possibly corrupt the computation of the cross-covariance term and thereby the overall track-to-track correlation and the resultant fused track estimate [1]. Correspondingly, it may be possible through IW attacks for example to create artificial violations of these conditions and the same type of degradation as would occur normally. When track fusion- based approaches are used, it is this combined and refined data that is used to drive the engagement decision and implementation software. It is the redundancy step in the reconnaissance functional area that provides a built in quality assurance function. Data from one source that does not correlate to the fused data from the other sources can be automatically flagged as suspect—unassociated data will be used according to whatever non-track hypotheses are in the association logic, e.g. FA's, new tracks, etc. Also, the addition of false targets through new or conventional EW techniques can create significant complication and error into this process, depending on its sophistication—i.e., if countermeasure techniques can create ambiguities in the association process by creating "atypical" data for which the algorithm would not likely have hypotheses, then such data would cause confusion in that (a) the overall likelihood of valid hypotheses would be lower, and (b) concerns would arise about the large amount of unassociated data.

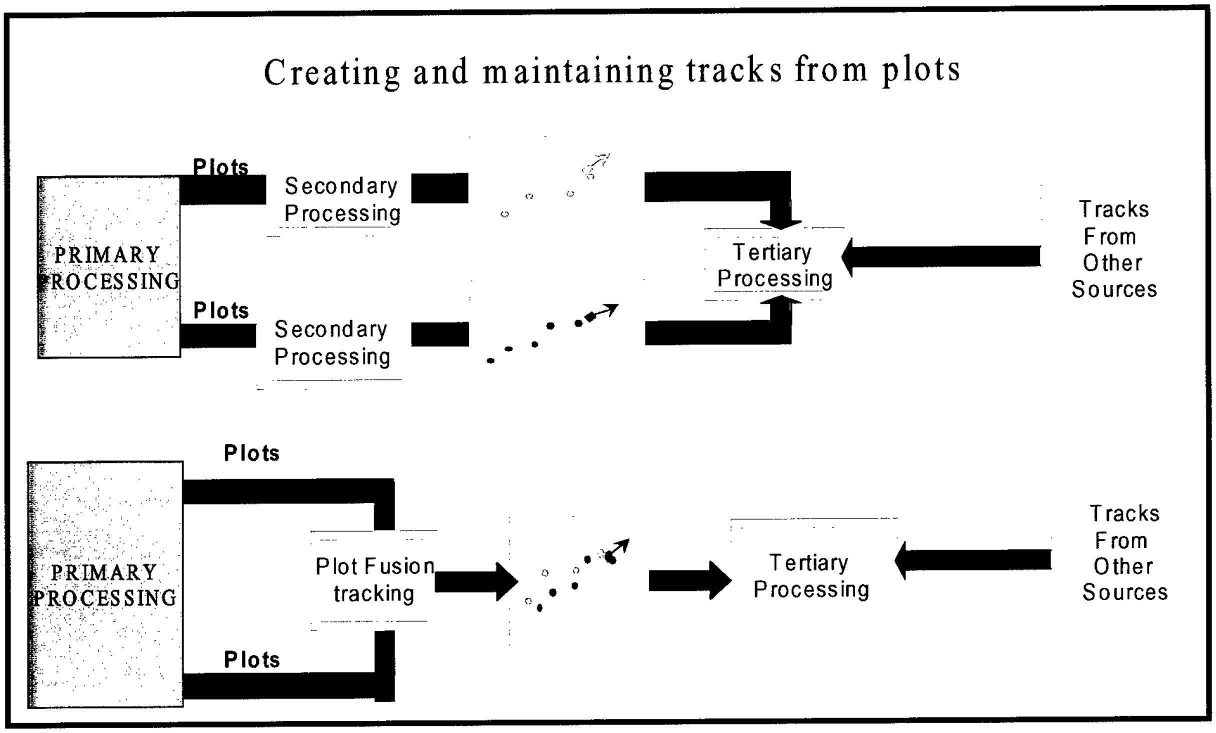

Figure B.6. Secondary Processing

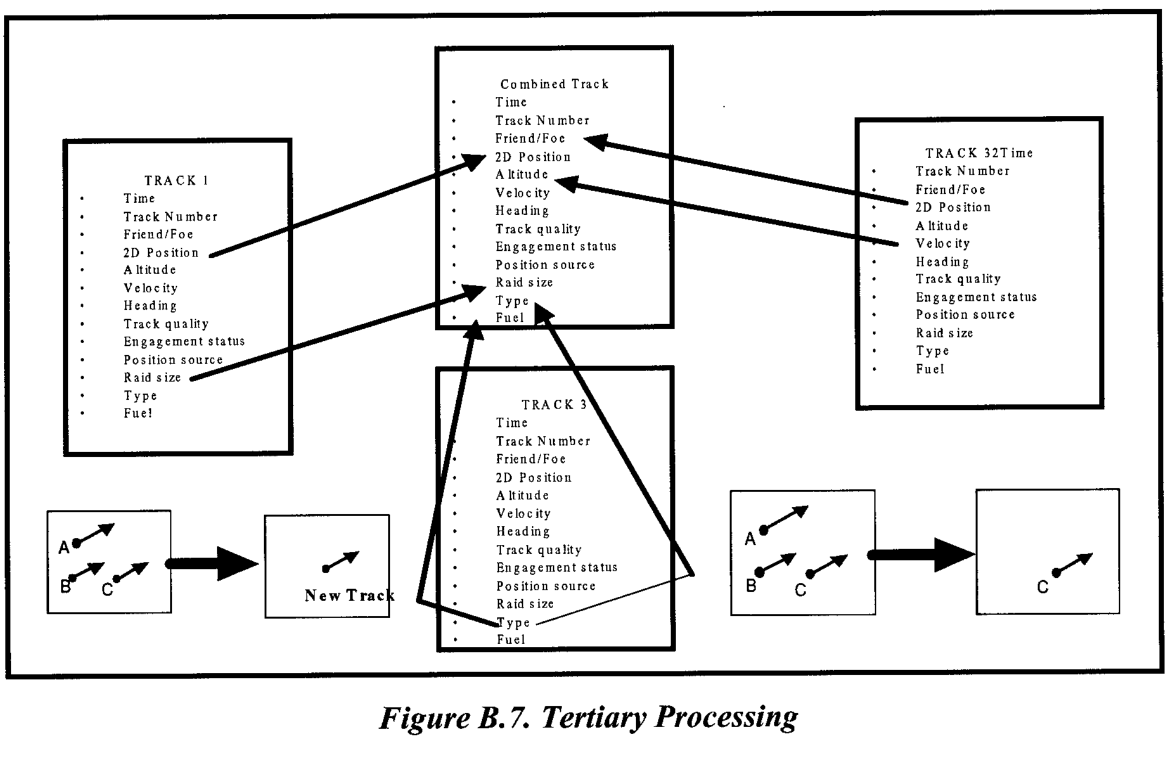

Figure B.7. Tertiary Processing

B.4 Targeting IADS Data Fusion With Countermeasures

The effectiveness of the IADS engagement decisions is directly correlated to the degree to which the estimated IADS air picture (i.e. the estimate produced by the automated fusion processes and, importantly, how this picture is perceived by the operator) represents the actual true air picture. Therefore the purpose of any countermeasure on the data fusion processes within a threat IADS is to either increase the gap between the perceived IADS air picture and actual truth or to purposely make the gap one that is advantageous to US force deployment. This is illustrated in Figure B.8, where the difference between the estimated + perceived air picture falls well short of the true air picture.

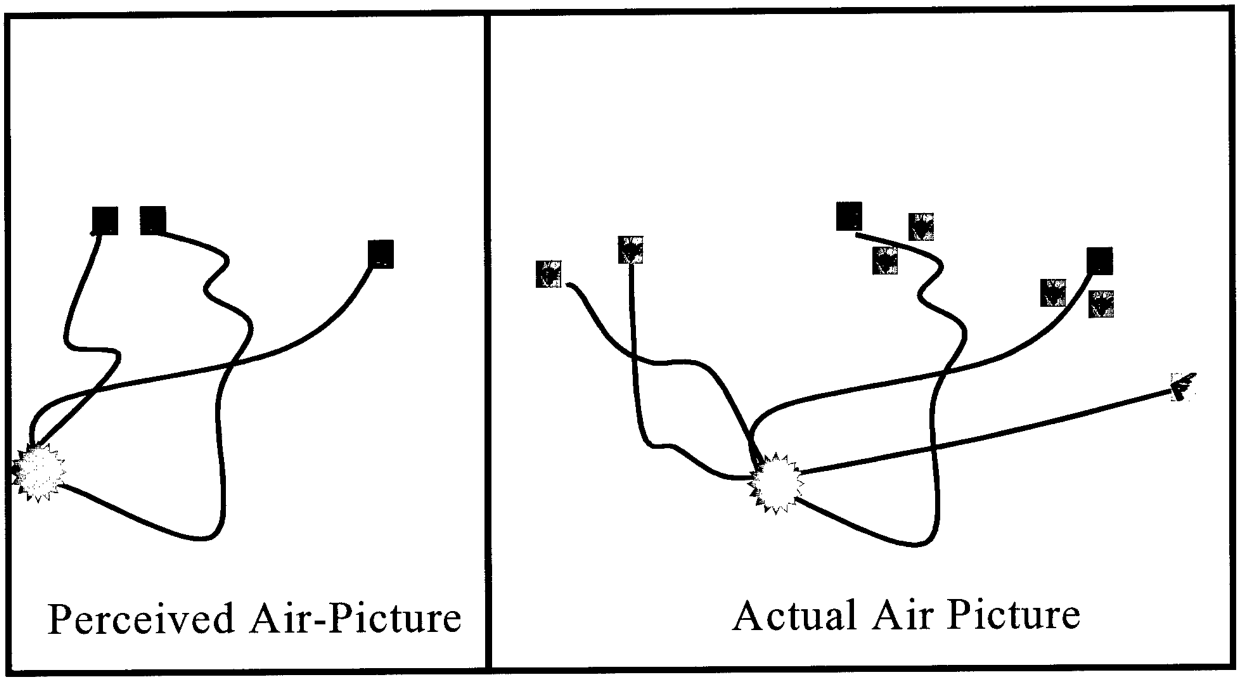

Figure B.8. Perceived vs. Actual Air Picture

To effectively use countermeasures to create, extend or control this gap between an IADS perceived and actual air picture, the concentration must be in the reconnaissance functional area. This is because it is through the reconnaissance area that the data flow to the command and weapons control areas is determined. It is this data flow or lack thereof that controls the decisions in the command and weapons control. The command and weapons control processes and algorithms must be understood in detail, because they provide the key to the actual effect of various combinations of reconnaissance data denial, distortion and deception that will be introduced into the reconnaissance functional area via countermeasures, but it is in the reconnaissance areas that the effect must be implemented. This is a very important comment—it says, in effect, that the marginal benefit of attacking the DF element of the adversarial IADS depends on how the DF "product" (tracks) is used by the C3 and FC functions; a sensitivity study should be done in this area after the attack modes against the hostile DF are defined. The next few sections will outline countermeasure concepts for each of the three basic processes within the IADS reconnaissance area.

B.4.1 Countermeasure Techniques and Primary Processing

Most convention countermeasure approaches attack the area of primary processing because it involves aspects of fundamental signal processing and detection that are well-known, well-studied areas of electrical engineering, and for which it is easier to estimate the types of techniques employed by an adversary. Said simply, it is an area that is well understood. However, as regards the informational aspects of such strategies, most of them are designed to work against a human operator and have not been explicitly optimized against an extractor (i.e. detector). That is, deceptive, denial or confusion strategies do operate directly against the extractor techniques but with an ultimate purpose of causing some level of error on the part of the operator.

The same ECM techniques that tend to overload and confuse a human operator (especially overload, via false, non synchronous targets) may not work against a computer algorithm if the overall process were automated. However, even the results of a highly-capable algorithm (one that can sort true and false targets well) may nevertheless produce an overwhelming array of targets to prosecute, if this were the case in the true air picture (i.e. that a large number of true and false targets were present)

Within the limit of reviewing some of the open literature, it seems that no *detection fusion* occurs in modern-day hostile IADS processing systems. While some searching on this point has been done, this should be checked because there is a wealth of information on the topic of detection fusion that one would presume the adversarial world is looking at—i.e. it has been well-studied by the fusion community (many papers, and a new book in 1998 [2]), and there is a big difference in attacking a system that has detection fusion processing vs. one that does not. For example fused detection performance can overcome the false-alarm limits of a single sensor, and offer the potential for new integrated design approaches to detection in multiple-sensor configurations.

Until the determination of whether detection fusion techniques are embodied in adversarial detection or extractor processing, one concept for employing countermeasure against the IADS primary processing component is to re-evaluate our current techniques and capabilities against a wide range of single-sensor automatic extractors. Techniques that do not work against a human operator such as standard noise jamming may work very well against extractors (i.e. in a strategy attacking the detectors directly). However, optimization of countermeasure against an extractor requires, that the exact algorithms within the extractor be known. Even in the single-sensor detection-processing area there are relatively new techniques being created and used in prototype systems if not yet operational systems. Thus, this is also an area requiring further research, to determine the possible employment of new detection processing techniques even for the single-sensor case. Methods such as quantized likelihood ratio (QLR) techniques that are used in single-sensor processing (and also used in detection fusion methods) are multiple-threshold approaches that rely on a quality bit being assigned to the different intervals between thresholds. This technique can be powerful in that local detection thresholds can be made very low, e.g., to detect low-signature targets either in or out of clutter.

B.4.2 Countermeasure Techniques and Secondary and Tertiary Processing

The impact of countermeasures on secondary processing has been virtually ignored until the present time. It is in the area of affecting secondary processing that the employment of countermeasures has new and significant implications. This is because the IADS does *not* engage or react to detections or contacts, but the intercept/weapon employment decisions are made on the basis of *track* data. The point here is that engagement decisions are serious decisions involving commitments of aircraft, threat to human life, etc—and thereby require support with the best information possible. Single plot points (the result of primary processing) simply do not contain adequate information to support an engagement decision, so corruption of those signals and data does create error but errors in perception, not necessarily and not directly affecting engagement decisions. Further, when thinking about the notion of engaging aircraft, it must be realized that the existence, current location, and kinematic behavior (and implied intent, etc) of true and false aircraft are all *virtual*—i.e. as estimated by the processing system in the overall track display produced by either secondary or tertiary processing, depending on the particular system. The operator has no sense of the true air picture; he only has the *estimated, virtual air picture* produced by the fusion process. The IADS transmits and engages tracks and the tracks are created and updated by the secondary processing component. The output of the secondary (or tertiary) processing component, whichever produces the composite estimated air picture, drives the logic that determines when and where engagements will take place. However, as for any such technique, optimization of countermeasures against a secondary processor also requires that exact or nearly-exact algorithms within the tracker be known.

Before discussing countermeasure strategies for secondary processing, a very brief review of post-detection-to-track estimation will be given, to assure consistency of terminology. After detections are determined (one could call these "valid" measurements in the sense of exceeding a detection threshold), these measurements are passed to Association and Correlation logic in which they:(1) are "scored" (this is usually a likelihood function) in the sense of gauging their "closeness" to an existing track (in effect to a predicted measurement for that track), and (2) they are "assigned" to a target, i.e. to a particular track estimation algorithm which has been running for a given track. This latter step is a combinatorial optimization process that considers the entire scan of measurements, either from one or many sensors, and the associated scores to optimally assign the measurements to tracks. Subsequent to these operations, the measurement is then used by the tracker (tracking algorithm) to propagate its estimates and predictions. The association/correlation step also usually involves what are called "gates" or spatio-temporal filters that would filter out or otherwise handle data well outside expected limits. However, this must be done very carefully and can be a rather complex logic unto itself. For instance, target maneuvers can create what might seem like an outlier measurement when in fact it is simply the maneuver causing a new sudden trend in the data; thus, tracker logic usually has some type of (possibly complex) maneuver gate logic included, even before the association step. If, on the basis of its score and the assignment process it is determined that the measurement does not associate to an existing track, logic must exist to somehow process this datum as perhaps a new track (i.e. to initiate a new target track) or as a false alarm to be discarded or otherwise processed, or some other logic to deal with such cases. Track initiation logic ranges from very simple and ad hoc to somewhat complex, and is based on the degree of a priori information on expected target complex specifications (number and type of targets etc). Similarly, there is today a large repertoire of estimation algorithms used for tracking, which range in sophistication and capability. If no measurements are received by the tracker (from the assignment logic) for some time, the track is usually dropped, unless the process is adaptive, wherein the tracker can call for adaptive sensor management to reacquire the target and continue the measurement stream.

The types of processing errors that can possibly be induced in exploiting the secondary processing software include:

1. Make selected data from primary processing unable to be processed (e.g. confound the gating or association processing)
2. Causing the track initiation process to consistently or selectively fail and restart (this would occur by defeating the assignment processor, causing many unassociated data and track fragmentation)
3. Confounding the initiation logic (how to do this would depend on knowledge of that logic)
4. Causing selected tracks to dis-associate from its aircraft (while feasible, this would be challenging to do selectively but in any case relies again on defeating the association logic)
5. Causing selected tracks to not be transmitted (this would be done by corrupting the track confirmation logic; see comments on track confirmation below)
6. Causing selected tracks to be dropped (multiple-scan corruption of the association process to break off the measurement stream)

By and large, as regards secondary and tertiary processing *for track estimation per se*, we have 2 cases: measurement fusion (MF) in the secondary case, and track fusion (TF) in the tertiary case; for each of these, we want to sketch out some ideas on how to corrupt their processing.

(MF) In the measurement fusion case, the creation of high levels of FA's can create considerable difficulty for the association component of the fusion process, and possibly overload the local-to-global node data link.

(MF) & (TF) If the inter-sensor communication patterns of hostile system sensors are known, it may be possible to fly routes or create synthetic data that cause data-stream latencies (out-of-order reports) to occur for given targets; while there are ways to overcome such effects, these effects typically cause degradation in tracking. Similarly, if the (presumed asynchronous) hostile scan and sampling patterns are known for each sensor, friendly trajectories (weaving in and out of scan patterns), whether real or synthetic, can be optimized to cause confusion in the track processing logic associated with track misses and in the formation of track scores for the asynchronous-sensor case. These errors add to the degradation of the fusion-based tracking process.

(MF) & (TF) One way to cause possibly serious tracking errors is to develop a method to induce systematic bias into these processes—bias effects cannot be easily or accurately removed by the estimation procedures unless the bias characteristics are known, which of course they would not be if they came from a suppression technique

(MF) & (TF) Experience on AWACS has shown that the major recon operator workload problem is manual track re-initiation. This is primarily a result of fragmentation of tracks due to severe and complex single and multiple-platform maneuvering. Data Fusion methods can aid in preventing these conditions but even modern US systems have difficulty with such cases. Sensitivity studies could also be done in this area, to investigate how particular types of platform dynamics (again, real or synthetic) can lead to confusion in fusion-based hostile recon/IADS systems. What is key, in terms of affecting hostile operator workload (making it high, and hopefully causing errors, etc), is to define those maneuver conditions which prevent or minimize the tracker from maintaining continuous tracks.

(MF) In the track fusion (tertiary) case, a critical design parameter is the track confirmation thresholding technique. This is the method, typically involving a "score" for the track that estimates its uncertainty (probability that a track represents a true target), according to which a local sensor decides to send the track estimate forward to the global fusion tracker to be included in the global filter's calculations. One aspect of this calculation is that the manner in which it is done can affect the time taken at the global tracker to form a corresponding track (presumably the track reported at the system level). Such effects are shown below in Figure B.9 (from [3]) where it can be seen that:

--any track confirmation scheme in support of track fusion is seriously delayed in comparison to the measurement or plot fusion case
--the results for high probability levels of confirmed tracks (say 0.9) can be considerably different

Note that the cited case was done for a hypothetical two-radar air defense system.

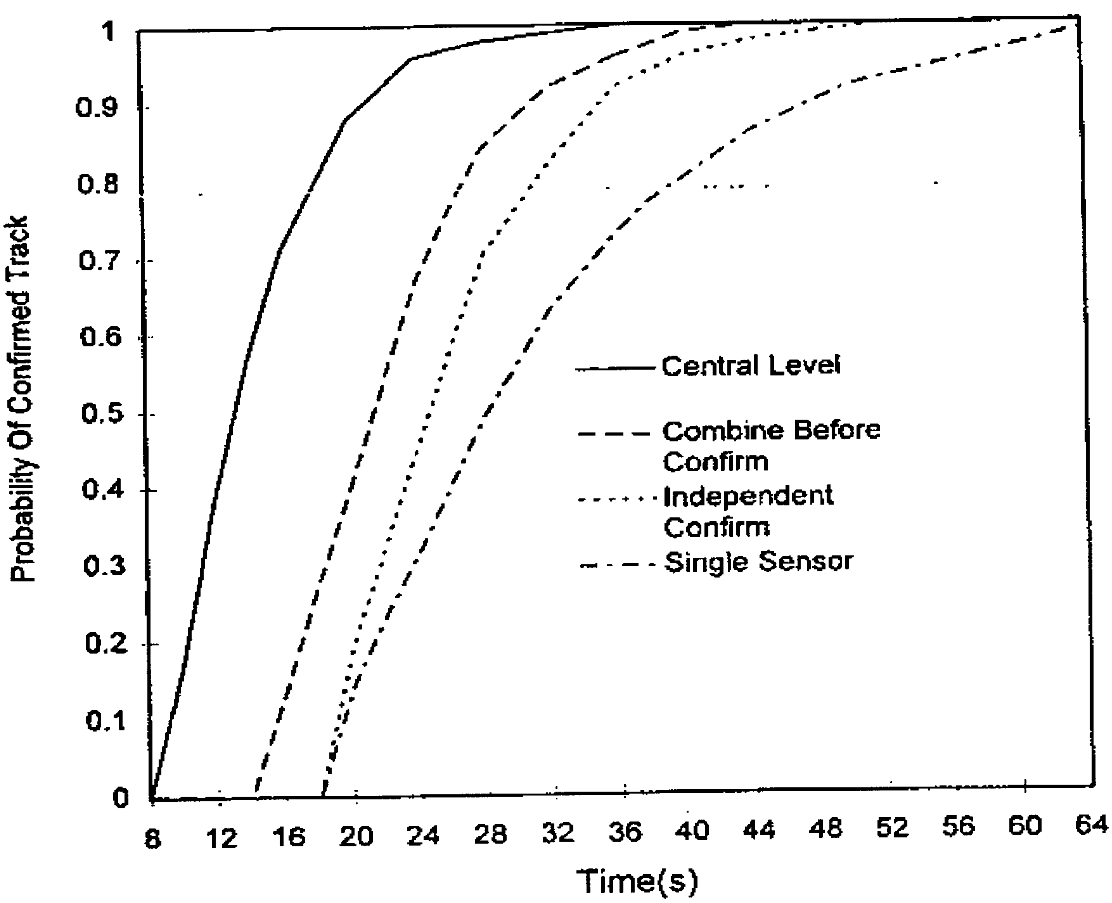

Figure B.9. Probability of Confirmed Track for Various Confirmation Logics (from [3])

The IADS is an *integrated* system; our attacks against it are usually performed by integrated strike packages. In the past our evaluation of the effectiveness of a given countermeasure concentrated on one versus one effectiveness. The redundancy of the IADS may well result in a targeted countermeasure approach having great success against a few IADS components with absolutely *no change* in IADS effectiveness. This often results from the IADS ability to fuse information across multiple sources and to eliminate bad information. The only way to attack tertiary processing is to consistently attack the secondary sources. This results in a requirement to have the effects of countermeasures distributed throughout the IADS, against all secondary-processing nodes.

B.4.3 Effects of Conventional Jamming on Data Fusion Processes

The effects of noise and repeater jamming on various IADS processing and decision-making stages are described in Table B. 1. The corresponding impacts on an IADS that employs Data Fusion processing operations can only be broadly characterized but any degradation in single-sensor processing or in target-specific processing will certainly have some type of degrading effect on resultant DF products (estimates). An important factor is, as always, whether the system being attacked with these methods has anticipated such effects and allows for the consequences in the overall logic. For example, if the false target characteristics generated by a repeater jammer are not accounted for in the association logic, various type of fusion degradation could occur to include degradation in accuracy or lost tracks.

B.4.4 Data Link Jamming

In the distributed data fusion environment of the modern IADS, it is perhaps clear that any corruption of the inter-nodal exchange of information will degrade the fusion results at some level. The purposes of data link jamming (and exploitation) are: to detect the enemy network's air picture and weapon assignment status (really an exploitation function), to deceive communications links by injecting false targets, and to deny communications links from passing data (tracks, commands, reports) between C2 nodes. False target injection has the usual saturation-related effects whereas denial actions have the effect of reducing the number of Secondary inputs to Tertiary processing. Figure B.10 gives a notional depiction of a typical data link structure for an IADS. In this figure, CP = command post and WCP = weapons command post.

Table B.2 depicts some of the usual flows of information along the links; this diagram confirms our assertions above that the basis of IADS operations revolves about track data. One other way to corrupt such a distributed tracking system is to inject redundant track-data-containing messages especially into the Tertiary node. If that node is in fact doing Track Fusion processing, then any redundant track estimates will corrupt that process at some level. This could be done by a technique that repeats the track data along a link from a Secondary processor to the Tertiary processor, presuming that the message-header information could be altered to prevent Tertiary from identifying the message as redundant.

Table B.1. Effects of Traditional Jamming Techniques on IADS Processing and Weapon Assignments

Traditional CM Technique	Targeted Process (Overt target)	Effect on Hostile IADS Primary Processing	Effect on Hostile IADS Secondary Processing	Effect on Hostile IADS Tertiary Processing	Effects on Weapon Assignment
Noise Jamming	Detection (Primary Processing)	•Target in clear region will be detected as normal. •Target in or near jamming will not be detected automatically without increasing the probability of false alarms. •Target in or near jamming may be extracted, but may require complex manual processing	•Target in clear region will be tracked as normal. •Target track entering jamming area will be tracked in a degraded mode. •Target originating in jamming area will not be initiated automatically without increasing the false track rate. •Target in or near jamming may be initiated manually.	•Targets in clear region will be tracked as normal. •If a confirmed source is being jammed, then another source will have to be used to track the target.	•Noise jamming may deny range of the target making SAM assignment and handover more difficult. •Noise jamming may highlight a priority target (or even a strike) causing the enemy to assign multiple assets.
Repeater Jamming (False Target Insertion)	Detection and Tracking (Primary and Secondary Processing)	•True targets will be detected and extracted as normal. •False targets will be extracted automatically if criteria are met: –Amplitude –Pulse width –Azimuthal width •False targets must move or they may be classified as fixed clutter by the extractor. •False targets must look like real targets on the radar video or the operator may override automatic extractor.	•True targets in clear region will be tracked as normal. •False targets will be tracked automatically if criteria are met. •False targets must move as normal tracks or they will be dropped. •False tracks interleaving with real tracks can confuse the secondary RDP algorithms. •A large number of false tracks can saturate the secondary RDP algorithms.	•A large number of false tracks can saturate the tertiary RDP algorithms. (This happens if the Secondary tracks are confirmed and passed on to Tertiary)	•Automatic and/or manual assignment of false tracks to SAMs and fighter •Increased acquisition time of true targets •Inefficient use of weapon resources •Saturation of weapon resources

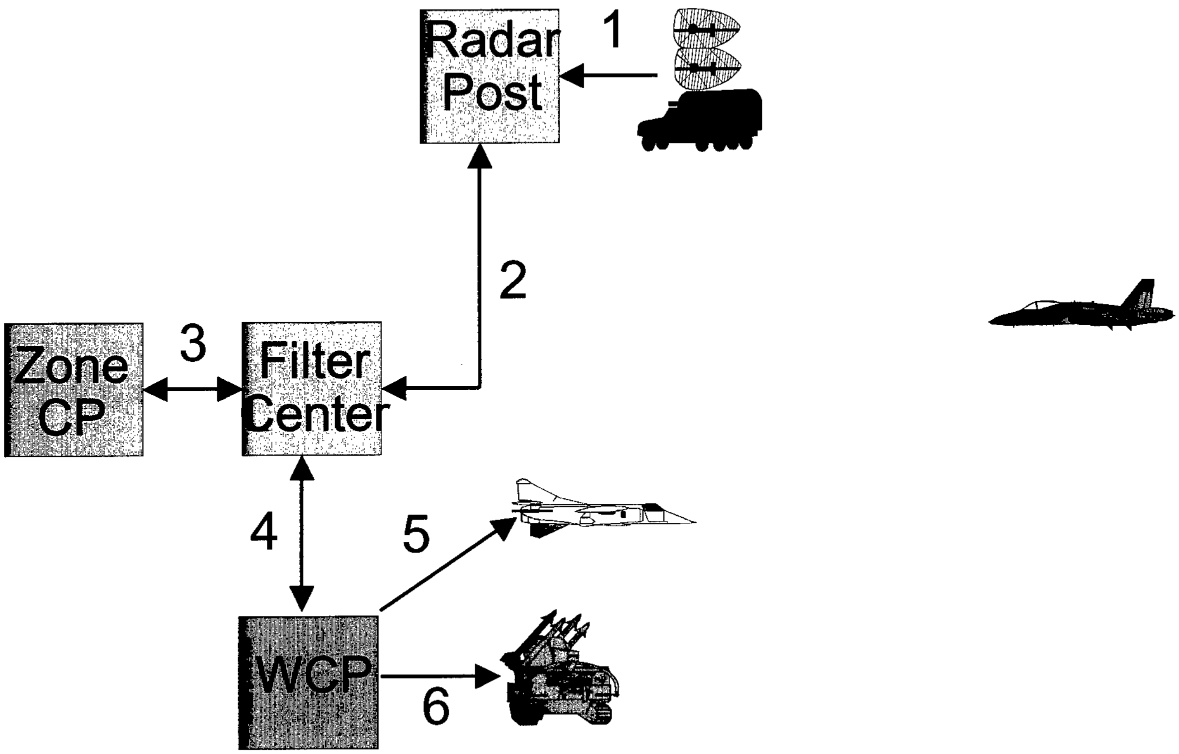

Figure B.10. Representative IADS Data Link Structure

Table B.2. Internodal Data Linking in IADS

Link	Node	Plots	Tracks	Track Management	Weapon Command	Comment
1	Radar to RP	X	X			May use land line
2	RP to FC	X	X	X		
3	FC to Zone		X	X		
4	FC to WCP		X		X	May use land line
5	WCP to AI		X		X	
6	WCP to SAMS		X		X	

B.5 Summary on Hostile IADS and the Role of Data Fusion

To attack the IADS we need to exploit our knowledge of both its overall automated processing logic, which aids in producing decision-supporting information, and also our knowledge of how this information is used in decision-making and in the overall C2 and engagement/kill-chain process. We can do this in part by attacking the data fusion processes within the IADS, which are important, perhaps even critical components of the overall information-creation process in the IADS system. Most modern or even relatively-modern IADS systems include some form of data fusion logic which offers new opportunities for exploitation by non-lethal means. As for any exploitation approach, what can be done depends on the breadth and depth of knowledge the friendly forces have about the targeted adversarial system.

The migration of the IADS internal data fusion processing operations from human to software processes provides an opportunity to re-focus the emphasis on non-lethal countermeasures. We can re-focus our efforts at exploiting the actual system software algorithms. This approach will allow the development of tactics, such as timed turning sequences that will result in dropped tracks, the insertion of messages to hide our true intentions, the stopping of selected messages to similarly deceive or selectively starve the software.

However, the real advantage of this approach is that it allows the countermeasure approach to be offensively rather than defensively minded. Denying information to minimize engagements is a defensive approach, but providing a mix of real and false information to a data fusion system that produces engagement environments favorable to our forces is an offensive approach. Rather than stem the flow of information, offensive strategies manipulate the data flow to have the software generate the information that leads the adversarial commander to make the decisions we want him to make.

While the US is in a leadership position with regard to data fusion technology and techniques, the US data fusion community has not studied the problem of hostile fusion process attack and exploitation very much. Most of the R&D has been focused on developing effective US intelligence, surveillance, and reconnaissance (ISR) systems, which are part of the defensive/awareness side of military policy, doctrine, and tactics. However, strategies and techniques to carry out data fusion-based countermeasures can indeed be conceptualized (as described herein), and could be developed into working prototype software for research and experimentation. The important issue of robustness (effectiveness versus degree of uncertainty or ignorance of the details of adversarial fusion techniques) of such data fusion-based countermeasure techniques would have to be studied parametrically, at least to some degree, since defining such robustness on strictly analytical grounds would be quite difficult due to the complexity of the overall fusion process on the one hand and due to the decoupled nature of that process (denying the ability to study inter-process effects with closed-form mathematics) on the other. However, the potential payoff is great, in the context as mentioned above of proactively redirecting adversarial decision-making to decisions favorable to blue force operations.

References

[1] Saha, R.K., "Effect of Common Process Noise on Two-Sensor Track Fusion", Jl. Of Guidance, Control, and Dynamics, 19, 4, Jly-Aug 1996.

[2] Varshney, P.K., "Distributed Detection and Data Fusion", Springer-Verlag, New York, 1997.

[3] Blackman, S. and Popoli, R., "Design and Analysis of Modern Tracking Systems", Artech House, Norwood, MA, 1999.